Challenges in Unstructured Mesh Generation for Practical and Efficient Computational Fluid Dynamics Simulations

Yasushi Ito[a,*]

[a]*Japan Aerospace Exploration Agency (JAXA), 6-13-1 Osawa, Mitaka, Tokyo 181-0015, Japan*

Abstract

An efficient and robust unstructured mesh generator, the Mixed-Element Grid Generator in 3 Dimensions (MEGG3D), has been developed to create good-quality meshes quickly and easily. MEGG3D has several key components for computational fluid dynamics (CFD) simulations, such as: 1) file compression with open-source data-compression utilities for faster file transfer; 2) three automatic local remeshing methods to modify existing hybrid volume meshes for additional components; 3) an easy-to-use hybrid surface mesh generation method to represent high-curvature surfaces with high-aspect-ratio quadrilaterals; and 4) multiple and suppressed marching direction methods to improve the quality of semi-structured elements in convex and concave corners, respectively. In this paper, those key components are introduced with practical examples.

Keywords: Unstructured mesh generation; hybrid mesh generation; file compression; remeshing; computational fluid dynamics; CFD

1. Introduction

Owing to the advances in computing hardware and computational algorithms, Computational Fluid Dynamics (CFD) simulations have been widely used for daily practical applications. Mesh generation is essential preprocessing in computational simulations, which is still sometimes a bottleneck of the overall simulations. Table 1 shows a brief comparison of mesh types widely used in computer-aided engineering. Structured meshes have been used for CFD simulation for a long time because of their computational efficiency. They are still essential for simulations requiring high-order spatial accuracy, while mesh generation is often troublesome and time-consuming for complex geometries.

Cartesian meshes have a long history since the introduction of computational simulations. From the view point of mesh generation, they are the best because they can be automatically created even for complex geometries with small geometrical defects, which must be repaired beforehand for the other mesh types. However, the treatment of walls sometimes becomes a problem to perform accurate simulations. Several methods have been proposed to resolve this geometrically and computationally, such as cut-cell methods [1], the building-cube method for handling large Cartesian meshes efficiently [2], and the immersed boundary method [3], and there is still a room for further improvements.

Unstructured meshes have been used for practical applications as a compromise approach because their computational efficiency is acceptable with computational algorithms diverted from those originally developed for the structured meshes, and today's computational resources. Isotropic tetrahedral meshes are used for inviscid flow simulations, and hybrid or anisotropic tetrahedral meshes for viscous flow

---

Presented at the International Workshop on Future of CFD and Aerospace Sciences, the RIKEN Advanced Institute for Computational Science, Kobe, Japan, April 23-25, 2012. NOTICE: this is the author's version of a work that was accepted for publication in *Computers & Fluids*. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in *Computers & Fluids*, Vol. 85, 2013, pp. 47-52, DOI: 10.1016/j.compfluid.2012.09.031.

* Corresponding author. Tel.: +81-50-3362-2894; fax: +81-422-40-1382.
E-mail address: ito.yasushi@jaxa.jp.

simulations to capture boundary layers efficiently with semi-structured elements or anisotropic, right-angled tetrahedra. The unstructured meshes can be created for complex geometries much more easily than the structured meshes, but there are still apparent disadvantages need to be considered and improved.

First, the file size of an unstructured mesh is much larger than that of a structured or a Cartesian mesh with a similar number of nodes because connectivity information for elements and boundary faces must be provided explicitly in the unstructured mesh file. This can be a problem when the files are transferred over slow network connection (this includes reading and writing the files on a network drive), computational simulations are performed on a cluster system that charges users based not only on CPU times but also on file size, or the files are stored for backup. Since the amount of the information in the unstructured mesh files cannot be reduced easily, an efficient method should be used to compress the information.

Second, turnaround time and user interventions should be reduced as much as possible in mesh generation for large simulations. However, it usually takes time to create large unstructured meshes. This becomes a critical issue when many meshes are needed in a process, such as a shape optimization process.

Third, high-aspect-ratio faces can be used in unstructured surface meshes to add an enough number of nodes only in an important direction (*e.g.*, the direction perpendicular to wing leading and trailing edges), but it is often not easy for users to create such faces. In typical mesh generation software, the users have to define a four-sided surface or something equivalent and specify node distributions on the boundary to create high-aspect-ratio faces using a structured mesh generation technique. A better method can be developed.

Forth, the quality of hybrid meshes should be improved for more accurate simulations. Advancing-layers type methods are widely used for semi-structured, near-field mesh generation, but they can produce poor-quality elements in convex and concave corners. Moreover, no semi-structured elements can be added around singular points.

We have tackled those disadvantages and have implemented several solutions in the Mixed-Element Grid Generator in 3 Dimensions (MEGG3D) [4-14]. To reduce the size of the unstructured mesh files, open-source file compression/decompression utilities have been used to directly output/input compressed files. To reduce turnaround time and user interventions, automatic local remeshing methods have been developed to create hybrid volume meshes for similar geometries. To create high-aspect-ratio faces easily, a new hybrid surface mesh generation method has been developed. To improve the quality of hybrid volume meshes at convex and concave corners, the multiple and suppressed marching direction methods have been developed. In this paper, we will introduce and demonstrate those methods separately in Sections 2 through 5. The conclusions and future work will be given in Section 6.

2. File Compression

The unstructured mesh files usually consist of two parts: a list of node coordinates and connectivity information for elements and boundary faces. The node coordinates are usually not very suitable to be compressed by lossless compression algorithms because their data is often close to random. The connectivity information contains integers only, which can be compressed efficiently.

There are three well-known, open-source, lossless compression utilities:

- zlib (http://zlib.net/)
- libbzip2 (http://bzip.org/)
- XZ Utils (http://tukaani.org/xz/)

Those can be linked with other software to directly read and write compressed files as well as to compress or decompress existing files. MEGG3D has linked with zlib and libbzip2, but currently does not with XZ Utils because they are written in C99, which is not supported by many C/C++ compilers. zlib and libbzip2 have their own features, and it is not difficult to use both of them because they have similar interface functions with similar arguments.

To demonstrate advantages of adopting compressed files, a hybrid volume mesh with 2.8 M nodes is chosen as an example. It is stored in a binary, element-based data format, and its size is 253 MB without compression and 99.4 MB as a gzipped file. The file size is significantly reduced by 60.7% as expected. Table 2 shows a comparison of CPU times for reading and writing the hybrid volume mesh on an internal drive and a network drives. One core of an Intel Core i7-2600 Processor is used. The maximum read and write speeds are measured using external software. The CPU times for reading the files depend on the read speeds from the drives and the performance of the PC because they include checking the range of node indices, and the reconstruction of face information to ensure the validity of the file. For the gzipped file, the performance is more important to decompress the data in it. From the network drive, the gzipped file is loaded much faster than the non-compressed file because of the smaller file size over the slow read speed. Interestingly, the CPU time for reading the gzipped file from the internal drive is also faster probably because of the high performance of the PC. The CPU times for writing the files mostly depend on the write speeds to the drives and the performance of the PC to compress the data. It takes more time to write the gzipped file than to read it, but this is reasonable because the data-compression algorithms generally need more time for compression.

Those results indicate that gzipped binary mesh files are the best choice for machines with a slow network drive. For high-performance machines, they might be preferred if the smaller size of the gzipped file is taken into account.

3. Automatic Local Remeshing

Two automatic local remeshing methods have been proposed for adding additional components to existing hybrid volume meshes [10, 12]. The first method [10] is for small additional components compared to the geometry in the baseline hybrid mesh, such as vortex generators and antennas. They are prepared as surface geometries, and can be added to anywhere on the baseline geometry. The second method [12] is for large additional components, such as a wing added to a fuselage. In this case, two or more overset-like hybrid meshes are merged to reduce the size of a remeshing domain around the additional components, which directly affects CPU time required.

Those methods have two remarkable advantages. First, hybrid meshes can be generated quickly and automatically from existing hybrid meshes when the additional components are moved and/or deformed. This enables practical shape optimization for complex geometries because many geometrical models and associated meshes are usually required during the process. Second, the updated hybrid meshes are the same as the original meshes except for the elements in the remeshing domains. The effect of the additional components can be evaluated more accurately. Acheson *et al*. show the importance of maintaining node distributions of volume meshes with and without a fuel vent in inviscid flow simulations as a practical industrial application for evaluating the effect of the vent [15].

One more automatic local remeshing method has recently been added to replace part of the surface mesh of a hybrid volume mesh. In the previous two methods, a Boolean union operation is used to create a water-tight surface from the baseline geometry and additional components [5]. It is robust for completely intersecting geometries, but cannot resolve tangential surface intersections as mentioned in [12]. However, such intersections are sometimes needed, for example, to optimize the shape of a nose radome. To explain the new method, Figure 1 illustrates a local remeshing process. In Figure 1a, a baseline hybrid mesh and a new surface mesh (dotted lines) are shown. There is one requirement to use the new local remeshing method: Nodes and edges of the baseline hybrid mesh corresponding to the boundary nodes and edges of the new surface mesh must exist in a certain tolerance (*e.g.*, black points in Figures 1a and 1b). Once those corresponding nodes and edges are identified, unnecessary part of the surface of the baseline hybrid mesh is automatically removed.

The remainder of the process is similar to the first local remeshing method [10]. Figure 1b shows a remeshing domain created around the new surface based on the distance, d_i, calculated at each node on it:

$$d_i = max(1.2 l_{i\max} \quad \alpha k)$$

$$k = \begin{cases} h_{tot} & \text{(for hybrid volume meshes)} \\ l_{s\max} & \text{(for tetrahedral meshes)} \end{cases}$$

where $l_{i\max}$ is the local maximum edge length of the baseline volume mesh, α is a parameter to control the size of the remeshing domain (currently set to 2), h_{tot} is the estimated height of the near-field mesh to be created on no-slip walls calculated from user-specified near-field mesh parameters, and $l_{s\max}$ is the maximum edge length of the new surface mesh. In Figure 1b, d is smaller than an actual value and constant because local maximum edge lengths of the baseline mesh are similar. In general cases, elements around the new surface mesh have to be removed based not only on the distance but also on element connectivity information [10]. If any of the semi-structured elements that are extruded from the same face (triangle or quadrilateral) on the boundary surface are removed, all of the elements are removed to make the near-field remeshing process easier (*e.g.*, hatched faces in Figure 1b). After that, the local remeshing is performed as shown in Figure 1c.

Figure 2 shows hybrid volume meshes created around the JAXA High-Lift Configuration Standard Model (JSM) with a pylon-mounted engine nacelle, leading-edge slats and trailing-edge flaps used in [10]. In this case, front and rear nacelle covers (Figure 2b) are added to the baseline hybrid mesh (Figures 2a and 2d). Since the covers are locally tangential to the nacelle, the Boolean union operation cannot be applied. Therefore, the new local remeshing method is needed. The resulting hybrid mesh is shown in Figures 2c and 2e.

4. Hybrid Surface Mesh Generation

A new hybrid surface mesh generation method has been proposed to create good-quality, layered quadrilateral faces around selected ridges even at concave corners with minimum user interventions [14]. It has two useful features. First, graphical user interface (GUI) is provided to distribute nodes on the ridges and to set parameters for creating the layered quadrilaterals (hereafter referred as *near-field mesh parameters*), combined with an automatic method for updating node distributions at concave corners. Second, the mesh generation algorithm is similar to the advancing-layers method [16], but high-quality quadrilaterals are automatically created at concave corners with special treatment, the so-called suppressed marching direction method. In the standard advancing-layers method, the connectivity of edges (faces in 3D) surrounding the remainder of the domain to be filled with elements is not allowed to be changed. This makes the method relatively easy to implement, while skewed semi-structured elements can be easily created, particularly at concave corners (Figure 3a). In the suppressed marching direction method, the connectivity can be changed at nodes on concave corners, where no marching directions are virtually defined, to avoid creating the skewed elements (Figure 3b).

To demonstrate those features, Figure 4 shows an example of a simple 1 x 1 square domain. Figure 4f is the mesh needs to be created, which has rectangular faces at the lower left corner. To create this mesh, only the near-field mesh parameters need to be specified on each of the lower and left ridges, in addition to the number of nodes on it. The square domain does not need to be divided into several patches to define a domain for triangles and those for quadrilaterals. In Figure 4a, ten nodes are placed on each ridge with equal spacings. In Figure 4b, the lower ridge is picked to set the near-filed mesh parameters. In this case, the maximum number of layers of 10, a stretching factor of 1.2 and a minimum spacing of 0.01 are set. The line segments emerged from the ridge indicate the total height of layered quadrilaterals to be added based on the parameters. In Figure 4c, the node spacings on the side ridges are automatically updated to adapt the change in the near-filed mesh parameters on the lower ridge. In Figures 4d and 4e, a similar change is happened on the left ridge.

5. Multiple and Suppressed Marching Direction Methods for Volume Mesh Generation

Typical advancing-layers type near-field mesh generation methods produce poor-quality semi-structured elements at convex and concave corners because they use a single normal extruded from each node on no-slip walls. To overcome this problem at convex corners, the multiple marching direction method [8] has been proposed (Figure 5). In addition, 3D surface models usually have a small number of singular points, where valid semi-structured elements cannot be placed using conventional hybrid volume mesh generation methods. With the multiple marching direction approach, good-quality semi-structured elements can be created around such points [8, 13].

To improve the quality of semi-structured elements at concave corners, another method, the suppressed marching direction method [14], has been proposed. This method can be considered as a 3D version of the hybrid surface mesh generation method described in Section 4. Figure 6 shows cross-sections of hybrid meshes at the wing-body junction of the ONERA M5 model near the trailing edge. Good-quality elements are desired in such regions because flow phenomena could be complicated because of side-of-body separation. Advancing-layers methods cannot create enough layers of good-quality semi-structured elements in the region, where the concave corner formed by the wing and the body and the trailing edge as a sharp convex corner meet, while the suppressed marching direction method can create more layers of good-quality hexahedra.

6. Conclusions and Future Work

To quickly and easily create good-quality unstructured meshes, several efficient methods have been implemented in MEGG3D. To reduce the size of mesh files, open-source data-compression utilities can be used. Compressed mesh files are particularly useful for machines with slow network drives. Three automatic local remeshing methods have been proposed to adjust existing hybrid volume meshes to additional components. Those methods are essential to perform shape optimization and to evaluate the effect of the additional components accurately. The hybrid surface mesh generation method is helpful to create high-aspect-ratio quadrilaterals around selected ridges easily. The multiple and suppressed marching direction methods have been proposed to improve the quality of semi-structured elements in convex and concave corners, respectively.

To further improve the effectiveness of unstructured meshes, mesh generation methods should be able to create large meshes more quickly. Although the automatic local remeshing methods have been proposed, baseline volume meshes still need to be created from scratch. To shorten turnaround time for mesh generation, an efficient parallel hybrid volume mesh generation method is required, which should start with surface meshes without any additional parameters for user friendliness. In addition, an automatic method is desired to create good-quality meshes from defective CAD files.

Acknowledgements

Most of the work described here was done in the Department of Mechanical Engineering, the University of Alabama at Birmingham, USA, and it was extended from the author's Ph.D. work under the guidance of Dr. Kazuhiro Nakahashi formerly of Tohoku University, Japan and currently of the Japan Aerospace Exploration Agency.

References

1. Aftosmis, M. J., Berger, M. J. and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952–960, DOI: 10.2514/2.464.
2. Nakahashi, K., "Building-Cube Method for Flow Problems with Broadband Characteristic Length," Computational Fluid Dynamics 2002, Armfeld, S., Morgan, P. and Srinivansan, K., Eds, Springer Verlag, Berlin, 2003, pp. 77-81.
3. Peskin, C. S., "The immersed boundary method," *Acta Numerica,* Vol. 11, 2002, pp. 479-517, DOI: 10.1017/S0962492902000077.
4. Ito, Y. and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490-496, DOI: 10.2514/2.1672.
5. Ito, Y. and Nakahashi, K., "Surface Triangulation for Polygonal Models Based on CAD Data," *International Journal for Numerical Methods in Fluids*, Vol. 39, Issue 1, 2002, pp. 75-96, DOI: 10.1002/fld.281.
6. Ito, Y. and Nakahashi, K., "Improvements in the Reliability and Quality of Unstructured Hybrid Mesh Generation," *International Journal for Numerical Methods in Fluids*, Vol. 45, Issue 1, 2004, pp. 79-108, DOI: 10.1002/fld.669.
7. Ito, Y., Shum, P. C., Shih, A. M., Soni, B. K. and Nakahashi, K., "Robust Generation of High-Quality Unstructured Meshes on Realistic Biomedical Geometry," *International Journal for Numerical Methods in Engineering*, Vol. 65, Issue 6, 2006, pp. 943-973, DOI: 10.1002/nme.1482.
8. Ito, Y., Shih, A. M., Soni, B. K. and Nakahashi, K., "Multiple Marching Direction Approach to Generate High Quality Hybrid Meshes," *AIAA Journal*, Vol. 45, No. 1, 2007, pp. 162-167, DOI: 10.2514/1.23260.
9. Ito, Y., Shih, A. M. and Soni, B. K., "Octree-Based Reasonable-Quality Hexahedral Mesh Generation Using a New Set of Refinement Templates," *International Journal for Numerical Methods in Engineering*, Vol. 77, Issue 13, 2009, pp. 1809-1833, DOI: 10.1002/nme.2470.
10. Ito, Y., Murayama, M., Yamamoto, K., Shih, A. M. and Soni, B. K., "Efficient Computational Fluid Dynamics Evaluation of Small Device Locations with Automatic Local Remeshing," *AIAA Journal*, Vol. 47, No. 5, 2009, pp. 1270-1276, DOI: 10.2514/1.40875.
11. Ito, Y., Shih, A. M., Koomullil, R. P., Kasmai, N., Jankun-Kelly, M. and Thompson, D., "Solution Adaptive Mesh Generation Using Feature-Aligned Embedded Surface Meshes," *AIAA Journal*, Vol. 47, No. 8, 2009, pp. 1879-1888, DOI: 10.2514/1.39378.

12. Ito, Y., Shih, A. M. and Soni, B. K., "Three Dimensional Automatic Local Remeshing for Two or More Hybrid Meshes," *International Journal for Numerical Methods in Fluids*, Vol. 66, Issue 12, 2011, pp. 1495-1505, DOI: 10.1002/fld.2324.
13. Ito, Y., Shih, A. M. and Soni, B. K., "Hybrid Mesh Generation with Embedded Surfaces Using a Multiple Marching Direction Approach," *International Journal for Numerical Methods in Fluids*, Vol. 67, Issue 1, 2011, pp. 1-7, DOI: 10.1002/fld.1962.
14. Ito, Y., Murayama, M., Yamamoto, K., Shih, A. M. and Soni, B. K., "Efficient Hybrid Surface and Volume Mesh Generation for Viscous Flow Simulations," AIAA Paper 2011-3539, 20th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, 2011.
15. Acheson, K., Kornegay, B. and Lau, H.-F., "Influence and Control of Unstructured Volume Meshes for Enhanced Fidelity Performance Predictions," AIAA Paper 2011-3984, 20th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, 2011.
16. Pirzadeh, S., "Unstructured Viscous Grid Generation by the Advancing-Layers Method," *AIAA Journal*, Vol. 32, No. 2, 1994, pp. 1735-1737, DOI: 10.2514/3.12167.

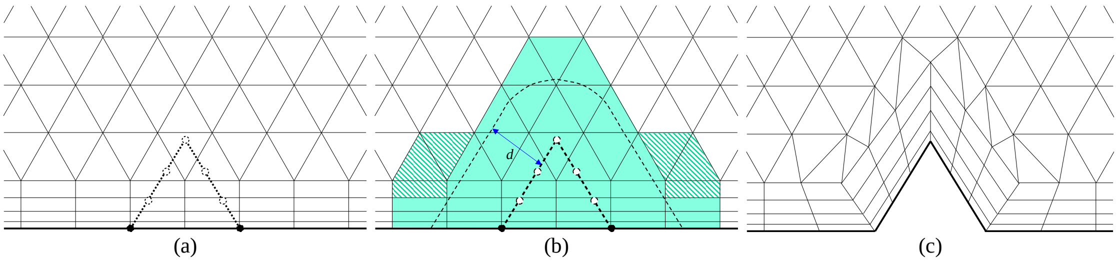


Figure 1. Local addition of new surface mesh to existing hybrid mesh in 2D: (a) baseline hybrid mesh (solid lines) and new surface mesh (dotted lines); (b) Remeshing domain defined around new surface mesh; (c) Modified hybrid mesh

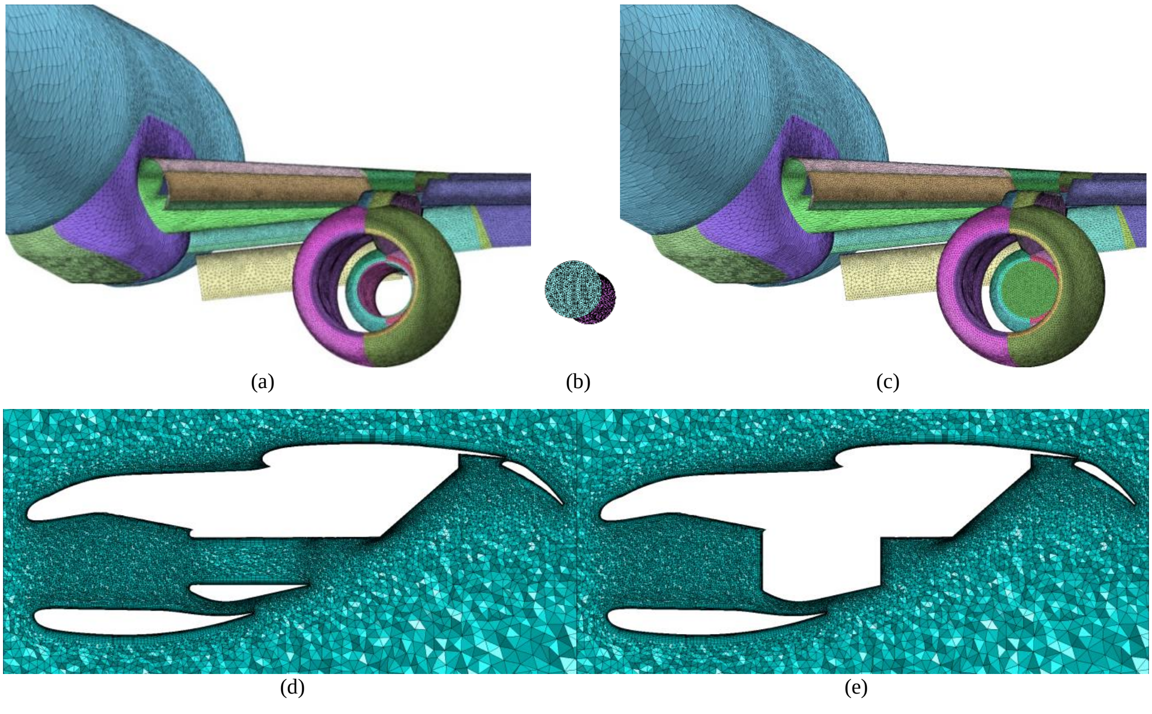


Figure 2. Addition of covers to flow-through nacelle: (a) Baseline hybrid volume mesh for JSM; (b) Nacelle covers given as surface mesh; (c) New geometry; Cross-sections of (d) baseline and (e) new hybrid volume meshes through center of nacelle

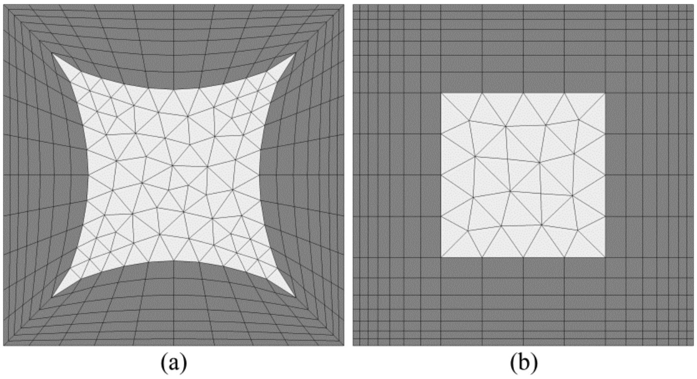


Figure 3. Hybrid surface mesh generation (a) without and (b) with the suppressed marching direction method: the shapes of quadrilaterals at the concave corners are greatly improved with the suppressed marching direction method

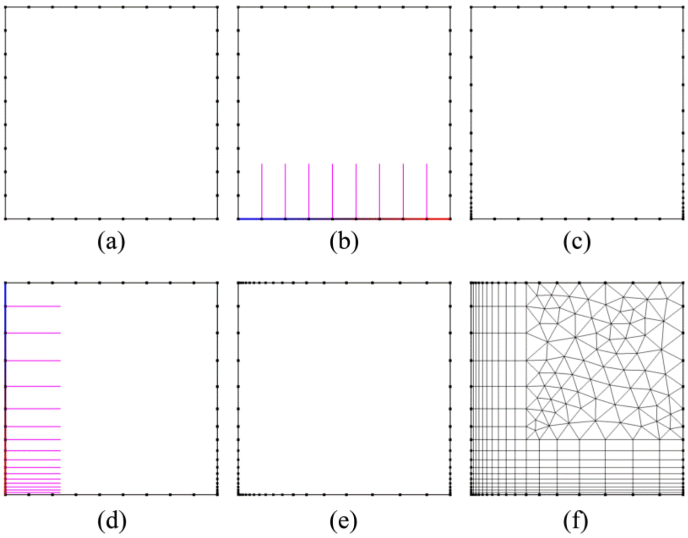


Figure 4. Node distribution on four ridges surrounding a square: (a) Ten nodes distributed on each ridge; (b) Lower ridge picked to set near-field mesh parameters—line segments representing the total height of layered quadrilaterals to be added; (c) Updated node distributions on the side ridges; (d) Left ridge picked; (e) Updated node distributions on the upper and lower ridges; (f) Surface mesh generated for reference

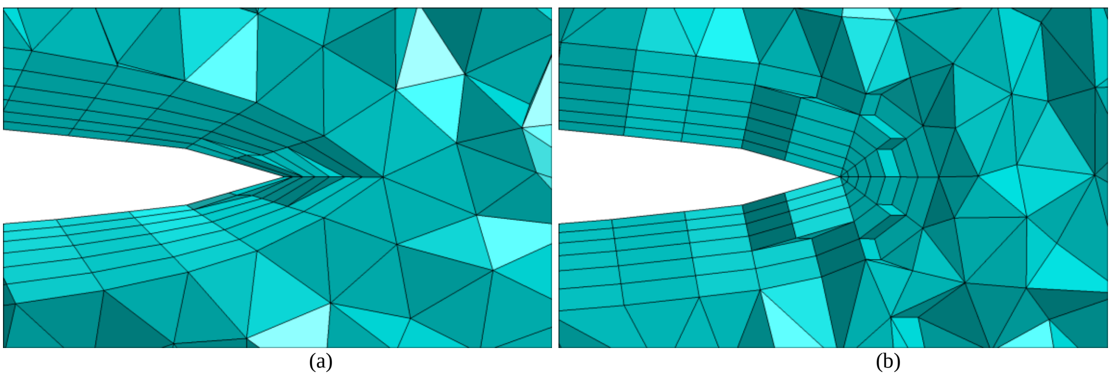
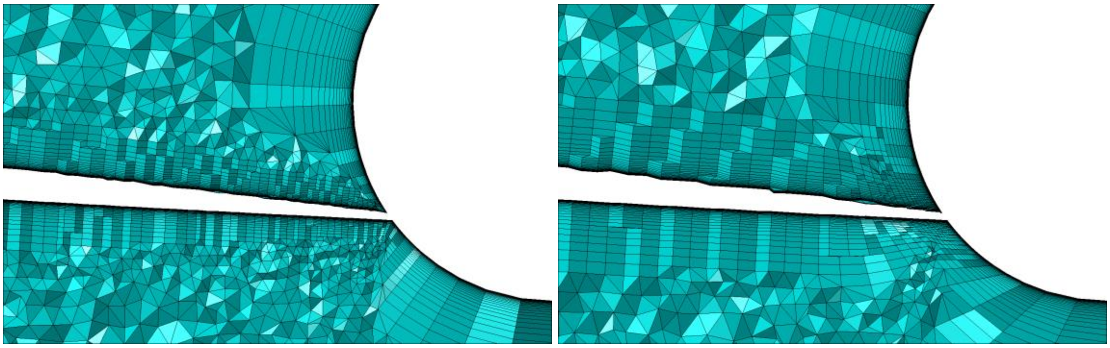


Figure 5. Cross-section of hybrid meshes at wing trailing edge (a) using single marching directions and (b) using multiple marching directions



(a) (b)

Figure 6. Cross-section of hybrid meshes at wing-body junction of ONERA M5 near trailing edge (a) using single marching directions and (b) using suppressed marching directions

Table 1. Comparison of mesh types

Mesh types	Computational efficiency	Meshing capability	Wall representation	Mesh file size
Structured	Excellent	Poor	Good	Small
Cartesian	Excellent	Excellent	?	Small
Unstructured	**Good**	**Good (Hybrid: Fair)**	**Good**	**Large**

Table 2. Comparison of CPU times for reading and writing a hybrid volume mesh with 2.8 M nodes

Drives	Max speed (MB/s)		CPU time for reading (s)		CPU time for writing (s)	
	Read	Write	No compress	Gzipped	No compress	Gzipped
Internal	400	400	8.3	**7.5**	**3.4**	24.4
Network	9.4	10.4	36.7	**15.6**	74.0	**39.6**