Automatic Triangular Surface Grid Generation on 3-D Surfaces Described in the Triangulated STL Format

Bong-Ju Lee*, and Byoungsoo Kim**

* ** Department of Aerospace Engineering, Chungnam National University, Daejeon, Korea
(Tel : +82-42-821-6686; E-mail: kbskbs@cnu.ac.kr)

Abstract: In this paper an automatic triangular surface grid generation method on general 3-D surfaces is proposed, where the 3-D surfaces are approximately defined in the STL format. The STL format is one of the standard formats for exchanging surface shape data, especially in rapid prototyping field, which replaces the original surface with a collection of triangulated surface segments. Triangulated surface data described in the STL format, which can be easily exported by most of CAD systems or 3-D modeling programs, replace the exact surface definition with air-tight triangular cells with some degree of approximation. The STL surface data, however, keeps the curvature information by putting a large number of smaller cells near the highly-curved region while using rather bigger cells near the relatively flat area.

The proposed algorithm for automatic surface grid generation is based on Advancing Front Method (AFM method). Starting with the surface data given in the STL format, feature edges which define the topology and geometry of the given surface are first identified by checking the angle between neighboring triangulated surface cells. When the feature edges are identified, the input surface is divided into a number of faces which are surrounded by chains of feature edges. Now that the feature edges and accordingly faces are identified, node points along those feature edges are distributed in an automatic fashion. Initial node distribution along the feature edges together with target cell size distribution along the surface is determined by checking the curvature distribution along the surface which can be estimated by the initial surface cell size distribution of the STL file with min/max cell sizes specified by the user. Those chains of initial nodes distributed along the feature edges are used as initial fronts, and the triangular surface grids are generated by using the AFM method. Projection of the newly-generated nodes onto the input STL surface following advancement of each front is performed to keep generated triangular cells to stay on the given surface data.

In this paper detailed description of the current approach is given, and results of automatic surface grid generation are demonstrated by using general 3-D surface shapes defined by CAD system and imported in the STL format.

Keywords: Surface Reconstruction, Grid Generation, CAD Surface Data, Advancing Front Method, STL Data

1. INTRODUCTION

As the Computational Fluid Dynamics improves its accuracy and extends the range of application, the turn-around time of CFD analysis becomes an important issue. In this regard grid generation is usually considered as a major bottleneck for a fast turn-around of numerical techniques in the engineering process because of its labor-intensive and error-prone nature. Therefore, development of automatic grid generation methods which can save time and labor in the process of grid generation has been one of important research fields in the CFD community these days.

In this paper an automatic triangular surface grid generation method on general 3-D surfaces is proposed, where the 3-D surfaces are approximately defined in the STL format. Initial node distribution and cell size is automatically controlled by using the surface curvature information estimated by using STL surface segments. Surface grids are generated one cell after another by advancing method which is basically based on the Advancing Front Method together with projection of newly-generated nodes onto the given surface to keep the nodes stay on the surface. [1]

2. AUTOMATIC SURFACE GRID GENERATION

2.1 CAD system and CAD-CFD interface

In recent years modeling of complex-shaped 3D objects are done using 3D CAD systems such as CATIA, and most of CAD systems can export surface shape data as a collection of triangular cells in the STL format. The STL format is one of the standard formats for the exchange of shape data, especially in rapid prototyping, which replaces the original surface with a triangulated surface. Triangulated surface data described in the STL format, which can be easily exported by most of CAD systems or 3-D modeling programs, replace the exact surface definition with air-tight triangular cells with some degree of approximation. The STL surface data, however, keeps the curvature information by putting a large number of smaller cells near the highly-curved region while using rather bigger cells near the relatively flat area. In this research, the developed program assumes the input data for surface description as STL format, and once the surface data are supplied to the program it generates unstructured surface grids on the given surface in an automated fashion. A triangulated surface representation of the RAH-66 Comanche helicopter surface imported in the STL format is shown in Fig. 1.

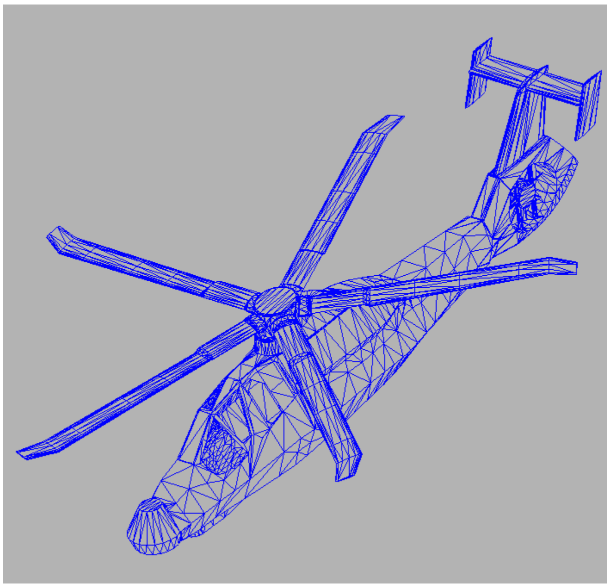
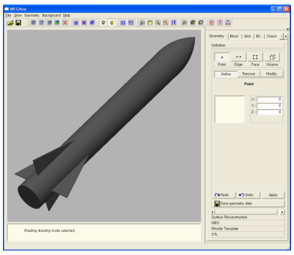


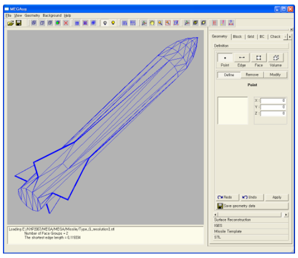
Fig. 1 Triangulated surface of RAH-66 helicopter

Surface definition data in the STL format describes the body shape by using only a collection of triangulated segments rather than using analytic functions and it does not provide with any of mass property and topology information. The fact that STL data does not have any mass property can be rather convenient for being used to generate grid system for flow calculation because internal mass property of the body is irrelevant to flow calculation any way and therefore it may be redundant and make the data file size bigger with little benefit. However, rack of topology information should be compensated somehow for the body surface to be used for grid generation, and the current approach used in this research to identify the topology information is by identifying feature nodes and feature edges, and more details of the method are described in the following section. Another issue in using STL data might be the resolution or accuracy of the geometry carried over in the STL format. When the shape is described in the analytic form, the geometry can be transferred exactly. However, once the body surface is triangulated and represented in the STL format, geometry approximation can not be avoided and any further detailed information finer than the resolution of triangular segments can not be restored.

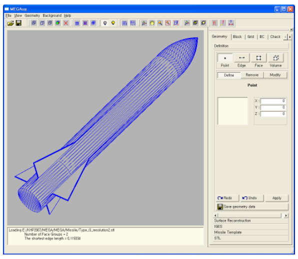
But this problem can be overcome by changing the resolution of STL representation by using CAD system-provided option, such that the resolution of STL triangles is finer enough than the target resolution of surface grid system. CATIA system also offers option to control the resolution of STL data, and Fig. 2(a) shows an object modeled in the CATIA system and other 3 figures show STL representation with three different resolutions.



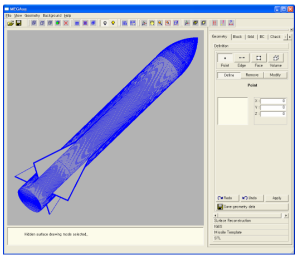
(a) Modeled object



(b) Coarse resolution



(c) Medium resolution



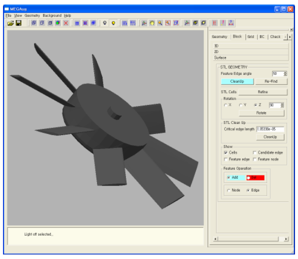
(d) Fine resolution

Fig. 2 Different resolution of STL representation

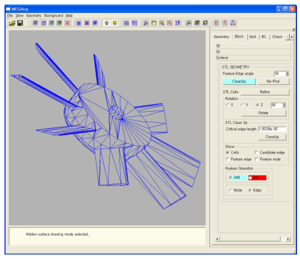
2.2 Surface grid generation algorithm

In this research a computer program is developed which can generate unstructured surface grid system in a fairly automatic fashion. After the user provides the program with the surface definition data in the STL format, the program performs automatic generation of unstructured triangular surface grids by following each of the steps described next and pictorially shown in Fig. 3(a)-(j).

1) Modeling: Modeling of an object is usually done by using CAD program such as CATIA. When the object is modeled, the geometry data is exported and the program reads the surface data input as a collection of triangular cells in the STL format. (Fig. 3a)
2) STL format: The STL surface data itself does not give any information about the topology of the object other than providing triangulated surface segments in an air-tight closure form. Therefore, the program needs to figure out the topology of the object by identifying feature edges and feature nodes which can tell how many edges, vertices and faces it is composed of. (Fig. 3b)



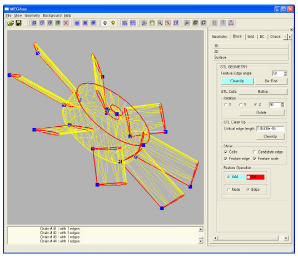
(a)



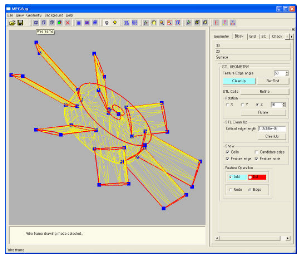
(b)

Fig. 3 Surface grid generation procedure(I)

3) Feature edges and nodes: Feature edges are automatically found by checking the angles between neighbor triangular cells. When other than 2 feature edges radiate from the same point, then it is considered as a feature node which is a vertex point of the body geometry. In the figure, feature edges are shown as thick lines, while feature nodes are shown as big dots. (Fig. 3c)
4) Curvature: The next step is to find high-curvature edges. When the geometry is described as an analytic function, then the curvature can be found analytically.[2] However, the surface geometry is replaced with a collection of triangular cells in the STL format, the curvature across an edge is approximated by the angle difference of two normal vectors of neighboring cells divided by the distance along the surface between two centers of those cells. In the figure, the leading edges of each blades have the highest curvatures, and those edges are also identified as thick lines. (Fig. 3d)



(c)

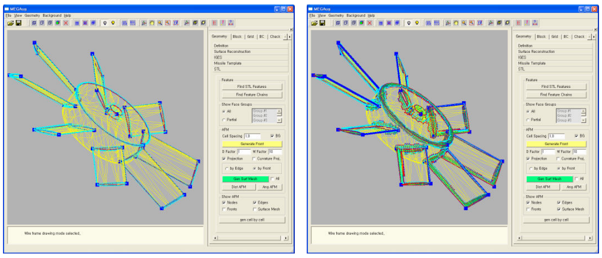


(d)

Fig. 3 Surface grid generation procedure (II)

5) Node distribution: When all of the feature edges and high-curvature edges are found, those edges are systematically grouped such that the whole object surface are divided into several faces which are bounded by a chain of the feature edges and high-curvature edges. When those faces are identified, boundary node points along those face edges are distributed. When calculating cell size distribution, user-specified minimum and maximum cell size is applied according to the curvature value such that the minimum size is applied to the highest-curvature edges. (Fig. 3e)
6) AFM method: This initial distribution of points along the edges is used as initial fronts for generating surface grids for each faces along the input surface by using the

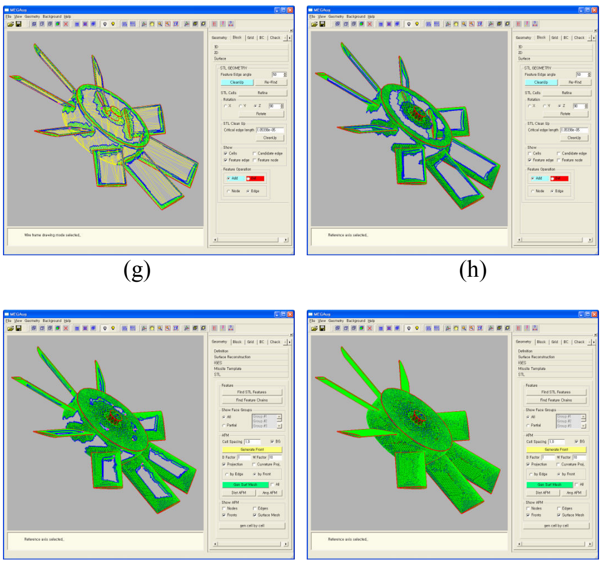
well-known Advancing Front Method (AFM). As can be seen in the figure, some fronts are advanced inward while some are advanced outward depending on whether they are outer boundary or inner boundary. (Fig. 3f)



(e) (f)

Fig. 3 Surface grid generation procedure (III)

7) Advancement along the surface: As the advancement of fronts along the 3-D surface proceeds, the un-traveled area surrounded by the fronts gets smaller and smaller, and when the area becomes 0 surface grid generation finishes successfully. If the surface of interest is flat, then the advancement of fronts occurs along that flat surface and the situation is relatively simple. But, for the generation of surface grid system, the advancement of fronts should be guided along the surfaces which are curved in general. If the advancement is not guided properly, then the resultant grid would not follow the given surface. In the current method this problem is overcome by projecting newly-generated node points onto the given STL surface segments following advancement of each front. (Fig. 3g-i)
8) When all of the given body surface are traveled by the fronts according to the method described above, then the surface grid generation is completed. (Fig. 3j)



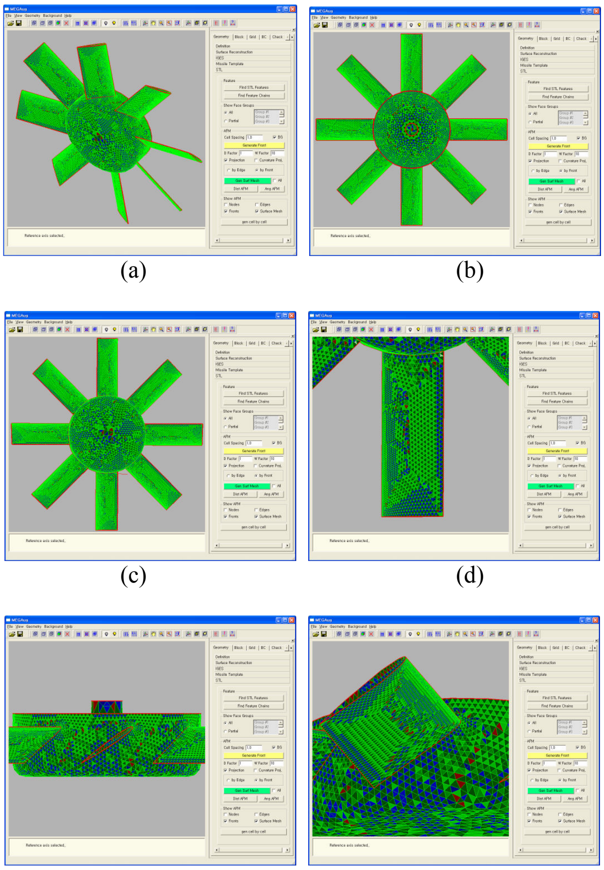
(g) (h)

(i) (j)

Fig. 3 Surface grid generation procedure (IV)

3. RESULTS

Fig. 4 shows the resultant surface grids of the previous example. The results show that surface grids are generated successfully and the cell size control according to the surface curvature leads non-uniform cell size distribution while each grid cell has good aspect ratio.



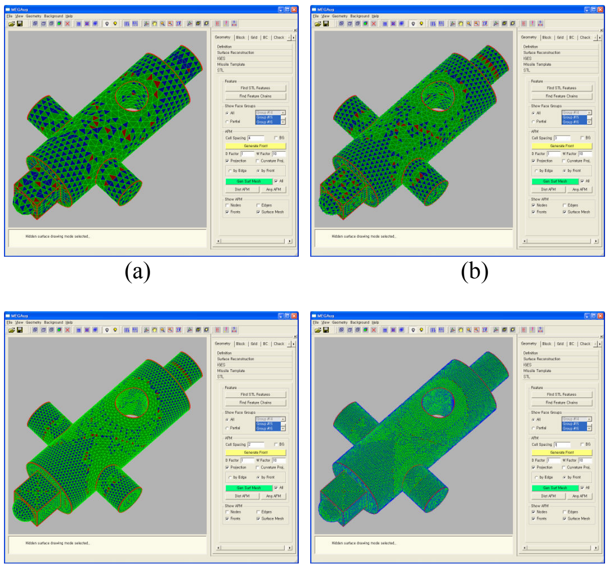
(a) (b)

(c) (d)

(e) (f)

Fig. 4 Resultant surface grids

Fig. 5 shows surface grids of different surface grid resolutions for a body with each other assuming basically constant grid spacing everywhere on the surface for each case.

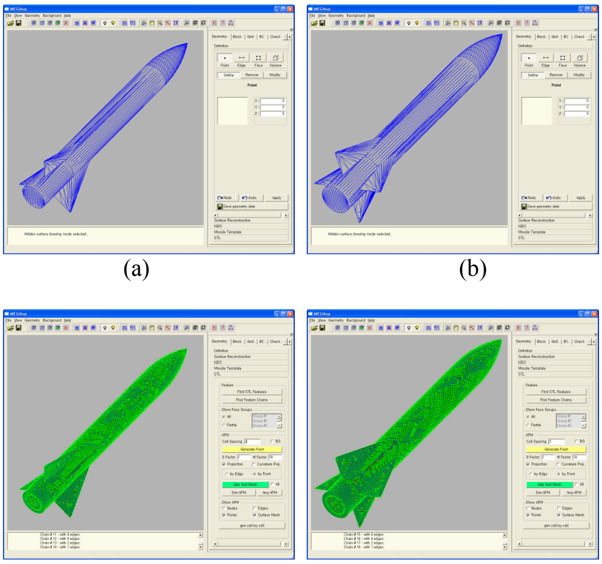


(a) (b)

(c) (d)

Fig. 5 Surface grids with different grid resolutions

Fig. 6 shows surface data in STL formats (a) for 3-fin missile and (b) 4-fin missile and the resultant surface grids on those missile-shaped bodies, respectively.



(a) (b) (c) (d)

Fig. 6 Examples for missile-shaped bodies

4. CONCLUSIONS

In this paper a computer program which enables an automatic surface grid generation of unstructured triangular grids for general 3D surface is described. The algorithm is basically based on the AFM method, while it advances fronts along the given body surface and it works successfully for geometrically degenerated cases too. And our efforts for surface reconstruction from three orthogonally-projected views of objects are also described.

Even though the program is still under development, it can be concluded that the program can be used to generate unstructured surface grids conveniently and almost automatically for general 3-D objects imported in the triangulated STL format. Further study for automatic grid sizing and inclusion of proper algorithm would make the program to be a very convenient tool for preprocessing for flow calculation and engineering.

In this paper an automatic triangular surface grid generation method on general 3-D surfaces is proposed, where the 3-D surfaces are approximately defined in the STL format. The algorithm is basically based on the AFM method augmented with projection of nodes to keep surface cells stay on the given body surface.

And it is shown that cell size control based on surface curvature information can make this kind of surface grid generation work in a fairly automatic fashion.

Nonetheless due to the nature of AFM method grid quality, especially in terms of cell skewness for regions where 2 confronting fronts merge together can be of bad quality. Further research to overcome this kind of problem and to acquire algorithm robustness needs to be done as the next step of current research efforts.

ACKNOWLEDGMENTS

This research has been supported by the KARI under KHP Dual-Use Component Development Program funded by the MOCIE.

REFERENCES

[1] S. Pizadeh, "Structured Background Grids for Generation of Unstructured Grids by Advancing-Front Method", AIAA Journal, Vol. 31, No. 2, pp. 257-265, AIAA, 1993.

[2] C. E. Pearson, Handbook of Applied Mathematics, Van Nostrand Reinhold Company, 1974