Visual Grid Quality Assessment for 3D Unstructured Meshes

Robert Haimes*

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology, Cambridge, MA 02139

Stuart D. Connell†

GE Research & Development Center
Schenectady NY 12301

Sabine A. Vermeersch‡

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology, Cambridge, MA 02139

Abstract

Areas of poor mesh can have a detrimental effect on the overall solution. With a structured mesh code grid quality assessment is a relatively straightforward task. Mesh plots on various coordinate planes may be made and visually inspected. Regions of poor mesh quality can then be identified and corrected. With unstructured meshes no such coordinate planes exist. Traditionally, when assessing a 3D unstructured mesh, a histogram of a cell quality measure has been used to determine if the grid is suitable for a solver. Although this procedure can inform the investigator where a region of poor quality mesh exists, it can do little to aid in understanding why the mesh is bad or how to improve the mesh. What is required is a method of visualizing the mesh in these regions.

This paper presents a method for displaying elements of a 3D mesh in an attempt to extract information about grid quality and topology. With this improved understanding, the input to the grid generator may be modified to produce a better mesh.

In addition these visual techniques can be used by the authors of mesh generation codes to further automate the meshing. Only by understanding where and what conditions cause poorly shaped elements can these situations be remedied or avoided.

The display of cell based quantities has additional uses other than mesh visualization. For example, if mesh refinement scheme is used where cells are tagged for refinement, then these cells may be displayed. This procedure can be used to help select adaptation criteria.

1 Introduction

Three-dimensional visualization techniques for flow solver post-processing have progressed greatly in the last few years. This progress stems from increased hardware capabilities and better algorithms. The introduction of graphics workstations, providing relatively affordable high-powered rendering and computational abilities, has been perhaps the greatest stimulus for the successes in scientific visualization. A wide variety of techniques are now available to many researchers.

This is not the case for viewing the results of grid generation codes. There has been almost no work, published or demonstrated, that will give the user a comprehensive view of the resultant mesh. (There has been some effort in graphical user interfaces for input to the mesh generators.)

Currently, the only effective way to view the interior of the mesh has been to look at how the grid intersects some user defined plane. If one tries to plot the entire grid, at best what is produced is a screen of 'spaghetti' or for fine cases a white display. Even for coarse grids, it is hard to see the trees (cells), and one only gets the forest. Interactive motion helps, but not much.

The work described in this paper has been built upon **Visual3** [1] [2] and is included in the current distribution. Because **Visual3** supports both structured and unstructured gridding, the methods discussed here work for most meshing techniques.

The example and much of the discussion in this paper will concentrate on unstructured gridding. This technology is relatively new for 3D CFD calculations and provides great

*Principal Research Engineer, Member AIAA
†Mechanical Engineer, Member AIAA
‡Graduate Student

flexibility in dealing with complex geometries with relatively small amounts of user interaction. Because of the topological freedoms (and the goal of hands-off grid generation) it becomes important for the user to assess the resultant mesh before spending many CPU hours in getting a solution that may be tarnished by small areas of the grid. In general, there has also been less experience with unstructured gridding than block and mutli-block structured codes.

It is important that the user have the ability to construct a grid quality measure that corresponds to the accuracy and stability of the flow solver's spatial and temporal discretization. A good discussion of many general mesh quality measures can be found in Woodward, Batina and Yang[3], but these are not directly tied to the solver's characteristics. **Visual3** allows the programmer to define any quality measure (and any number of these 'cell based scalar' fields) in a manner that is similar to the **Visual3** definitions of nodal based scalar and vector fields.

There are five popular methods for generating unstructured meshes. The first is the tetrahedration of a structured grid. This method has been used by Dawes[4] and others to mesh complex geometric topologies. It will not be discussed in that the cell shapes are controlled by the structured mesh. The second technique is Delaunay tetrahedration as used primarily by Baker[5]. This meshes a 'cloud' of existing nodes into tetrahedra that are well formed and as close to equilateral as possible. Modifications to the Delaunay method are required at boundaries to ensure that faces in the boundary triangulation exist in the final mesh. These modifications can lead to some poor quality tetrahedra in boundary regions.

The third method used for producing unstructured grids is the advancing front scheme used by Peraire[6] and others. Unlike the second method, the surface tetrahedra are usually well formed, though some 'bad' tetrahedra may be produced away from boundaries where the fronts merge.

The fourth method, popular in structural mechanics, is the octree approach, Sheppard[7]. Here the domain is recursively subdivided into a series of octants until some level of geometry definition is achieved. These octants are subsequently divided into tetrahedra. With this approach the mesh quality tends to deteriorate as the boundaries are approached. Mesh smoothing is used to alleviate this problem.

A new method that is gaining popularity is Steiner triangulation, Weatherill[8]. This method begins with a triangulation of the boundaries. These boundary facets are then tetrahedralized to form an initial mesh. Interior points are now recursively added in regions where the mesh quality is poor. This technique has the promise of producing meshes of any required quality.

2 Techniques

The new data type 'cell based scalar' has been added to **Visual3**. This is simply scalar information defined in each element of the computational domain.

Unlike the nodal scalar visualization, Gouraud-shading is not used. All the surfaces of an individual cell are drawn with the same color. This is because the entire cell has the same value. The color of the cell is based on the value of the cell scalar and the associated color map. Also, cell visualization is always thresholded by the cell scalar limits, that is, only those cells within the limits are plotted. Therefore it is usually important to histogram the elements to set a valid range before drawing.

The selected cells are only rendered into the 3D window. (See Haimes and Darmofal [9] for a discussion of **Visual3**'s dimensional windowing graphical user interface.) The elements can be opaque or transparent, and each cell's edges can be outlined. The rendered cells can be shaded in grey or a color that represents the value of the cell based scalar. And, the color map used to render the cell function can be different from the normal scalar map.

This plotting differs from 'Element Clouds' [10] which also plots the entire element, in that the cell retains it's size and is not displaying the nodal scalar field. 'Element Clouds' work well for very coarse CFD problems or structural analysis where the number of elements is usually less that 40,000. But this technique can not highlight areas of the gridding, and only show the general mesh topology.

2.1 Histogramming

This is the traditional method [3] for addressing grid quality assessment. It can give the investigator a statistical sense of the cell quality throughout the entire domain.

It is always useful to perform histogramming before initiating the following techniques. This allows the setting of restrictive limits so that only poor quality elements are drawn.

2.2 Surface Cells

The cells that are within the cell scalar limits and have a face on the 'active' surface are displayed in the 3D window. This should be used if the grid was generated with Delaunay or Octree approach to inspect the cells that touch the surface. Also grid quality measures can be constructed that reflect the accuracy and stability of the solver's boundary conditions.

This option is extremely useful for CFD applications where the key to a good solution is a high quality mesh near solid boundaries (usually where the largest gradients will exist). The solvers can usually handle poor quality cells far from boundaries where the flow is essentially one dimensional.

The regions where the elements don't pass the criteria may be regridded. In the case of Delaunay tetrahedration, more nodes can be added on the body so that the surface gets a finer definition. For the advancing front method, either the bounding surface grids will have to be re-triangulated or the spacing parameters of the background mesh in that region be adjusted. With the octree approaches further mesh smoothing could be employed in an attempt to improve the quality.

2.3 Volume Cells

All cells that are within the cell scalar limits are drawn in the 3D window. This is useful in inspecting the cells in the volume produced by the advancing front method. It may be possible to remove bad cells by adjusting spacing parameters of the background mesh in that area. Some bad cells may always be produced as the fronts collapse upon themselves. In this case a grid post-processor may have to be written that removes the nodes that are causing the problem elements and re-tetrahedrate (in a Delaunay manner) the resultant hole. The other methods (octree and Delaunay tetrahedration) tend to produce high quality meshes away from boundaries.

The level of interactivity can be severely impacted by using Volume Cells (based on the total number of cells plotted). For this reason, it is advisable to adjust the cell scalar limits to an appropriate and restrictive range before initiating this option. Also, to further enhance performance, some pre-processing is performed by **Visual3** if the cell connectivity data is available. This calculation determines if a cell to be drawn is completely surrounded by cells that will also be plotted. If so, the cell is not plotted. The user will not notice this unless clipping is used to cut into the data, then the hollows will be seen!

2.4 Cut Cells

The cells that are intersected by a cut (either planar, programmed or iso-surface) and are within the cell scalar limits are rendered in the 3D window. This option allows the visual inspection and correlation of the grid quality with the resultant solution quality. The user can answer questions such as; "Is this anomalous feature in the flow a reflection of the physics or do to poor gridding?" This could be done by displaying an iso-surface of entropy generation rate (for an Euler calculation) and displaying all cells which cut this surface. If there is a correlation between high entropy generation and poor grid quality the likely culprit is the grid rather than some flow feature.

When invoking Cut Cells, it is useful to draw the element using transparency. This way the underlying scalar field information can be seen through the cell data.

Another useful visulalization technique involves Cut Cells. 'Cobble-stone' cut surfaces allow the viewing of slices through the mesh by setting the cell scalar limits so no thresholding is performed. The investigator only sees all the elements that intersect the cut when the cells are plotted opaque. A complete sense of the mesh may be obtained by slowly moving (scanning) the cut.

3 Application

These visual grid inspection techniques are applied to the grid generation and flow over Lockheed's F117A stealth fighter. The reasons for the choice of this aircraft in the examples are that the overall geometry is simple (therefore defining the body surface is relatively easy) and the visual results are less obscured.

3.1 Grid Generation

The advancing front method is used for the volume discretization. First a program creates a triangular mesh on the boundaries of the three dimensional domain. These boundaries are specified by the user with a surface definition file. In this file, the user defines support points and intersection segments of the surfaces which make up the body. Every surface is represented by four support points which determine the orientation and location of the plane. Intersection segments are the intersection lines between two adjacent surfaces. They are defined by two support points. The user also has the ability to specify the spatial distribution of the mesh parameters which are interpolated from a background grid made up of tetrahedral elements. The spatial parameters define the spacing and the stretching of the tetrahedral elements in specific areas of the mesh. The surface mesh is stored in a front file.

Another program uses the surface definition file, the background file and front file to create tetrahedra using the advancing wave front method in the 3D domain between the specified boundaries and creates the final grid file.

3.2 Surface definition

In order to write the surface definition file, the spatial coordinates for the aircraft need to be specified. A 1/32 scale model of the F117 is used to generate the surface intersection points. It was assumed that the complete aircraft consists of 78 flat surfaces. The engine inlet and outlet were plugged. Each facet is defined by at least three node points. The spatial coordinates for each of the corner points are measured by using a milling machine and a micrometer. The set up is shown schematically in Figure 1.

Since the F117 is a symmetric airplane, only half of the necessary nodes are measured. The location of a total of 38 points is determined and then projected about the symmetry plane. Figure 2 shows the simple geometry description.

Because the data for the geometry of the F117A are measured manually, there are small errors involved in the spatial coordinates of each of the measured points. When defining the intersection segments and the support points for the surfaces, the four nodes which represent facet may not be co-planar. The surface triangulator corrects this by calculating an average surface and projecting the four nodes on the new surface. Figure 3 shows a grid created by the surface triangulator.

The final grid consists of 4326 triangles that bound the domain. A total of 9396 nodes support 48316 tetrahedra.

3.3 Body Examination

A variety of cell quality measures exist. One useful metric is cell aspect ratio. This is the ratio of radii of the circumscribed sphere to the inscribed sphere. This ratio is then normalized so that an equilateral tetrahedra has an aspect ratio of one.

The grid quality measure used in this example to display poorly formed cells is skewness, calculated in degrees. For this paper skewness is defined as the maximum angle that a face normal deviates from the vector between the node of the tetrahedron not on the face and the centroid of the face. A value of zero degrees would indicate an equilateral tetrahedron. A value close to 90 degrees would indicate a cell far from equilateral. These distorted tetrahedra have three forms: needle, cap and sliver [11].

Figure 4 displays all the cells with a skewness value between 80 and 90 degrees which touch the surface. Because the advancing front method was used, few cells touching the aircraft body should be poorly shaped. Only 4 cells have a skewness greater than 88 degrees. In general, they are in regions where surface facets intersect at oblique angles and there has been a poor match between the spacing parameter and the surface triangulation. Figure 4 also shows the histogram (from the 1D window) as well as the color key for skewness.

3.4 Volume Examination

Again skewness is used to display all the cells in the volume with values greater than 88 degrees. Figure 5 displays the image produced from **Visual3** for 2 different orientations. It should be noted that most of these cells are so flat that the edges 'bleed' through the faces. This causes the 'bow-tie' effect seen with the cell outlines.

In general, these cells are produced from the collapsing fronts or a spacing parameter that changes too rapidly. As seen in the histogram of figure 4 there are enough of these cells to cause concern to most flow solvers.

3.5 Solution to the 3D Euler equations

The method used to solve the governing equations is a Galerkin finite element discretization method. A four stage Runge-Kutta time-stepping scheme is used to obtain temporal discretization. Artificial dissipation is controlled by second and fourth difference terms. A pressure switch determines the amount of dissipation added. The second difference damping is turned on at sharp edges to enforce the Kutta Condition.

Figure 6 displays a planar slice through the volume. Entropy is painted on the cut along with contour lines. Cells with skewness of greater than 86 degrees that intersect the slice are drawn gray and transparent. Note the contours at the upper-right portion of the core. This shows the solver's difficulty in maintaining a smooth solution through a high gradient region with poor quality cells.

3.6 Adaptation

To illustrate another use for the displaying of cell based variables, an adaptive solution scheme is considered. In order to capture the important flow features, one level of adaptation by mesh point embedding is done. Entropy is used for the adaptation parameter so that mesh points are added in regions where flow features have created entropy. In this case the generation of vortices is the mechanism. Figure 7 displays the region selected for adaptation. The criteria used is the cell's average entropy.

It is important to point out that if adaptation is done by mesh enrichment where a tetrahedral cell is refined into 8 smaller cells, then no amount of adaptation will improve cells which are of poor quality. The new cells are of comparable quality to their parent.

4 Conclusions

The visual techniques described in this paper present methods for displaying elements of a 3D mesh that have user defined properties, grid quality metrics. With the characteristics of the flow solver (including the boundary conditions used) and the visual information extracted from the grid topology, the input to the grid generator can be modified to produce a better mesh.

If the ultimate goal in grid generation is 'hands-off' meshing, then the techniques describe here must be used by the people that write the grid generators. Only by understanding how poorly defined elements are created can these situations be corrected.

And finally, visual grid inspection can useful in selecting adaptation criteria (and levels) for nodal mesh enrichment.

Acknowledgments

The authors would like to thank Jaime Peraire of Imperial College for supplying the grid generator and Dave Modiano, a student at MIT, for the solver. In addition the first author wishes to thank the General Electric Corporate Research Center for their financial support.

References

[1] R. Haimes and M. Giles. VISUAL3: Interactive Unsteady Unstructured 3D Visualization. AIAA Paper 91-0794, 1991.

[2] M. B. Giles and R. Haimes. Advanced Interactive Visualization for CFD. *Computing Systems in Engineering*, 1(1):51–62, 1990.

[3] Batina J. T. Woodard, P. R. and H. T. Y. Yang. Quality Assessment of Two- and Three-Dimensional Unstructured Meshes and Validation of an Upwind Euler Flow Solver. AIAA Paper 92-0444, 1992.

[4] W. N. Dawes. The Extension of a Solution Adaptive 3D Navier Stokes Solver towards Geometries of Arbitrary Complexity. ASME 92-GT-363, 1992.

[5] T. J. Baker. Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delauney Triangulation. *Engineering with Computers*, 5:161–175, 1989.

[6] J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and O. Zienkiewicw. Finite Element Euler Computations in Three Dimensions. *International Journal for Numerical Methods in Engineering*, 26:2135–2159, 1988.

[7] M. S. Sheppard and Georges M. K. Automatic Three Dimensional Mesh Generation by the Finite Octree Technique. *International Journal for Numerical Methods in Engineering*, 32:709–739, 1991.

[8] N. P. Weatherill and O. Hassan. Efficient Three Dimensional Grid Generation Using the Delaunay Triangulation. First European Fluid Dynamics Conference Brussels, September 1992.

[9] R. Haimes and D. Darmofal. Visualization in Computational Fluid Dynamics: A Case Study. IEEE Computer Society, Visualization '91, October 1991.

[10] D. Darmofal and R. Haimes. Visual Feature Identification for 3-D Data Sets. AIAA Paper 91-1583, 1991.

[11] M. Bern and D. Eppstein. Mesh Generation and Optimal Triangulation. CSL-92-1, Xerox Corporation, Palo Alto Research Center, March 1992.

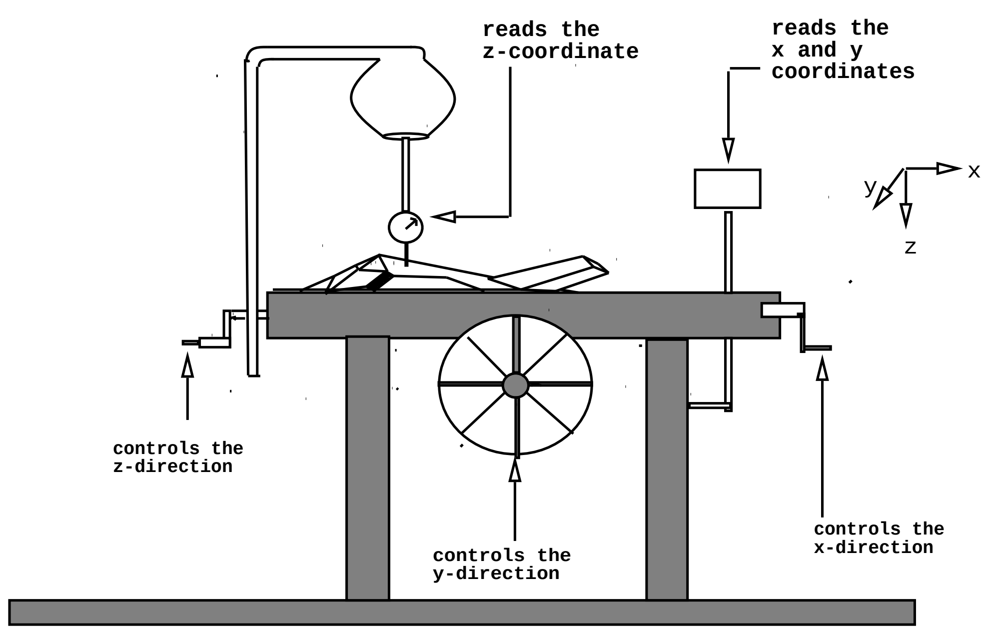

Figure 1: Acquisition equipment for spatial coordinates

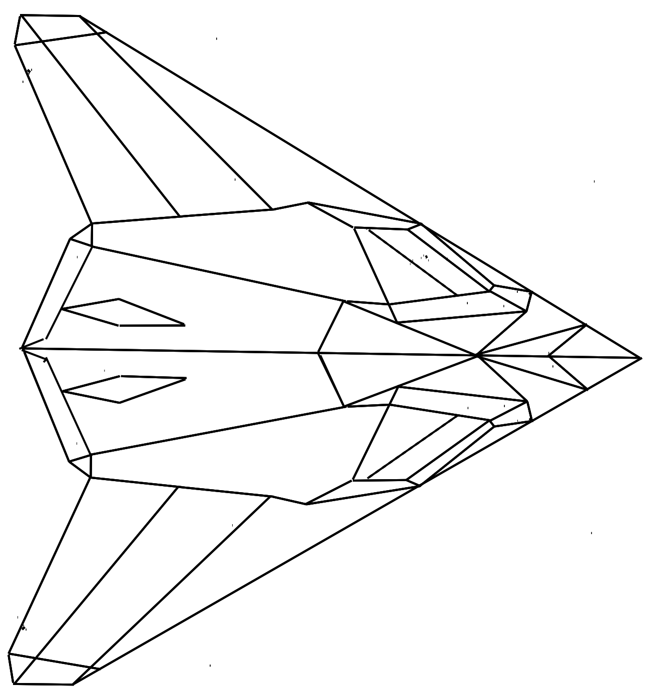

Figure 2: Geometry definition of the F117A

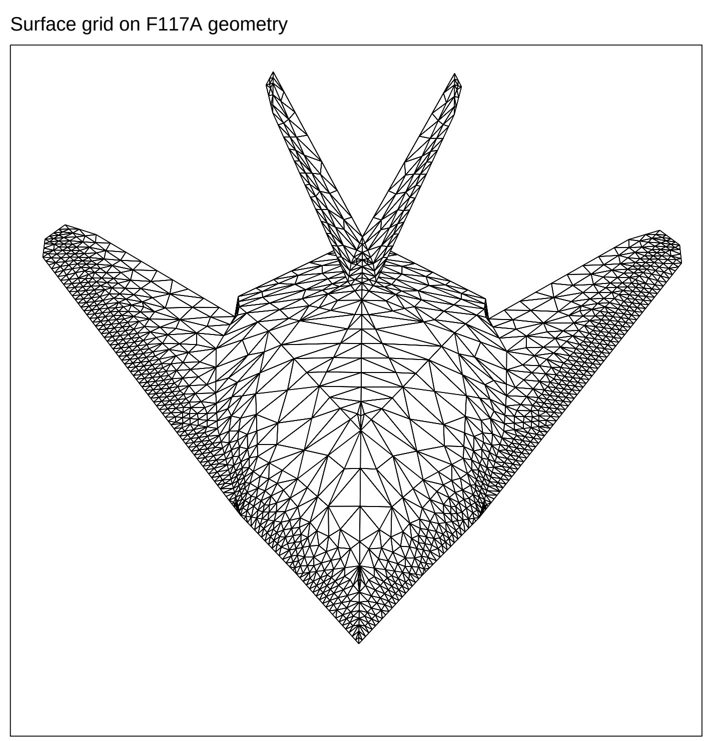

Figure 3: Surface grid

Figure 4: F117 showing surface cells

Figure 5: F117 showing volume cells

Figure 6: F117 showing transparent cut cells against a planar cut

Figure 7: F117 showing cells to be adapted