

AIAA 2000-2245

OVERSET STRUCTURED HYPERBOLIC GRID GENERATION ON TRIANGULATED SURFACES

William M. Chan
ELORET, Army/NASA Rotorcraft Division,
NASA Ames Research Center,
Moffett Field, California 94035

Fluids 2000
19-22 June, 2000 / Denver, CO

**For permission to copy or republish, contact the American Institute of Aeronautics and Astronautics
1801 Alexander Bell Drive, Suite 500, Reston, VA 20191–4344**

AIAA 2000-2245

OVERSET STRUCTURED HYPERBOLIC GRID GENERATION ON TRIANGULATED SURFACES

William M. Chan *
ELORET, Army/NASA Rotorcraft Division,
NASA Ames Research Center,
Moffett Field, California 94035

Abstract

A scheme is presented for the creation of overset structured grids on a surface geometry described by triangles. Two types of surface features are identified and preserved: seam curves and seam points. Seam curves lie along surface discontinuities and high curvature contours while seam points are points with non-unique normals on the surface such as corner and apex points. Methods are described for extracting these surface features on a triangulated surface. Other surface curve creation schemes on a triangulation involving cutting-plane intersection and stencil-walks are also presented. A hyperbolic marching method is then employed to generate structured surface grids from the surface curves and seam points over the triangulated geometry. Examples are shown for overset grids created for several complex surfaces including parts of a rotorcraft, a space vehicle, and a triceratops. With the aid of a graphical interface, overset surface grid generation time for a moderately complex geometry can typically be accomplished in about half a day.

1. Introduction

The efficient design and analysis of aerodynamic vehicles require rapid modelling of the surface geometry to create computational grids suitable for numerical simulations. Ideally, surface grids should be generated directly on the CAD representation of the geometry. Unfortunately, such tools are not yet conveniently available for creating structured surface grids that follow the Chimera overset gridding philosophy.[1] In the past, overset surface grid generators[2,3] utilized a multiple panel network representation of the geometry as input, where each panel network is a rectangular array of points that defines a set of bilinear quadrilaterals. Conversion from CAD data to multiple panel network format requires manual effort and can be a time consuming process. Tools that have been used for this procedure include GRIDGEN,[4] ICEMCFD,[5] and others. Clearly, a faster method to reach the CAD data needs to be found.

The ideal approach is to have independent library routines that any surface grid generator can access to perform tasks like point projection and surface normal interrogations on the CAD model. While such library routines do exist as part of commercial CAD packages, independent interfaces of this kind are not readily available. The CAPRI[6] application programming interface (API) provides interrogation routines for solid models created from Pro-Engineer, CATIA, and a few other commercial CAD packages, but the respective commercial licences are required to use the software. A next best option from direct CAD model interfaces is to employ a surface geometry format that can be derived quickly from CAD data, while keeping a high fidelity representation of the true surface geometry. Surface triangulations are excellent candidates for this purpose.

Many CAD packages already offer output in various surface triangulation formats. Moreover, a rapid and essentially automatic method to create high quality surface triangulations is available in the CART3D[7,8] software utilizing CAPRI. The maximum distance between a triangular face and the CAD model, as well as certain grid quality constraints, are prescribable to within a specified tolerance.

This paper describes a scheme that takes advantage of this easily available geometry format for creating structured surface grids. While the multiple panel network surface description has the advantage that small gaps and overlaps are allowed between neighboring panel networks, this is also a drawback since automatic algorithms for domain decomposition are difficult to achieve.[9] Since all face connectivities are known in a surface triangulation, construction of automatic algorithms may become a more tractable problem.

The Chimera overset grid approach for flow simulations using structured grids has been successfully applied to many complex configurations.[10–14] The overset surface grid generators[2,3] mentioned above employ hyperbolic grid generation methods which have proved to be efficient and convenient for the task. Resulting grids are of high quality, nearly orthogonal, and

*Senior Research Scientist, ELORET, Senior Member AIAA

This paper is a work of the U.S. Government and is not subject to copyright protection in the United States.2000

each is created by marching from a single specified initial curve. With a multiple panel network surface description, such initial curves are usually produced by extracting a subset of a network in the structured index space. With an unstructured surface triangulation, more sophisticated surface curve creation methods are needed. Moreover, such initial curves should also capture important geometric features on the surface. These problems are addressed in Section 2 of this paper. Hyperbolic grid generation methods from surface curves and corners are described in Section 3. In particular, a projection scheme for triangulated surfaces is presented. Several test cases are shown in Section 4 to illustrate the capability of the new scheme. Concluding remarks are given in Section 5.

2. Surface Feature Capturing

The first step in surface grid generation is to decompose the surface geometry into multiple domains suitable for the creation of surface grids. One of the main advantages of the overset grid approach is that such domains can overlap arbitrarily. The boundaries of these domains, and hence the boundaries of the surface grids, are allowed to float freely, making it easier to produce high quality almost orthogonal grids. However, constraints must be applied to the grid lines so that surface features of interest are properly captured. Typically, such surface features are captured by one or more of the grid boundaries. This suggests that the problems of surface domain decomposition and surface feature capturing are closely connected.

2.1. Surface Triangulation Data Format

The schemes presented in this paper are based on an indexed surface triangulation format described below. This format can also be easily converted from other triangulation formats. The triangulation is assumed to consist of a set of non-duplicated vertices. Connectivity between the vertices is prescribed by the indices of the three vertices that make up a triangular face. This information is stored in a face-vertex list. It is also assumed that none of the faces intersect each other and that the surface geometry is trimmed, i.e., only the outer mold line (wetted surface) is described. Each face of the triangulation also carries a tag which is an identifier for the component in which the face belongs. For example, a face in an airplane geometry may belong to the fuselage, wing, pylon or nacelle component. Such a triangulation is conveniently described by the CART3D[7] or FAST[15] formats.

From a given face-vertex list, the following data structures are built so that geometric operations needed for surface feature extraction can be conveniently performed.

Vertex-face list - the indices of the faces surrounding each vertex.

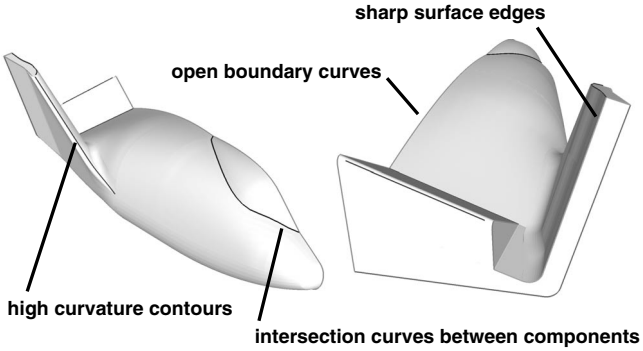


Fig. 1: X-38 Crew Return Vehicle example showing various types of seam curves.

Vertex-edge list - the indices of the edges surrounding each vertex.

Edge-vertex list - the indices of the two vertices connected to each edge.

Edge-face list - the indices of the two faces next to each edge. For edges on an open boundary, there is only one neighboring face.

The vertex-face list and vertex-edge list are stored as 1-D arrays. A pointer array is used to point to the first face and edge for each vertex on the vertex-face and vertex-edge list arrays, respectively.

2.2. Surface Feature Extraction

For a geometry described by any format, two types of surface features may be identified: seam curves and seam points. Seam curves can be classified into the following four types: (1) curves along surface discontinuities (e.g., wing trailing edge), (2) intersection curves between components (e.g., wing/fuselage junction), (3) high curvature contours (e.g., wing leading edge), and (4) open boundary curves (symmetry plane curve for a half-body). Seam points are singular points of the surface with non-unique normals and can be of two types: (1) a corner which is an intersection point of multiple seam curves, and (2) the apex point of a locally cone-like geometry with smooth cross section. Examples of these surface features are shown in Figures 1 and 2.

Detection and extraction methods for the above surface features are given below for a surface geometry described by a triangulation.

Surface Discontinuity Curves

Surface discontinuity curves are created by automatically concatenating a set of flagged edges on the surface triangulation. Using the edge-face list, an edge is flagged if the normals of the two adjacent faces deviate by more than a user specified threshold. Such a scheme is appropriate for capturing sharp discontinuities on the surface but is unable to detect high curvature regions with rounded turns.

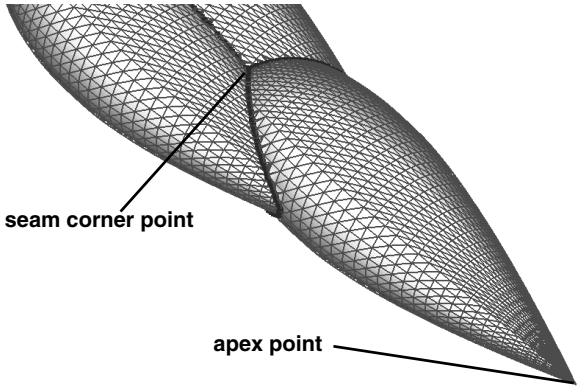


Fig. 2: Intersected teardrops example showing corner and apex points.

Component Intersection Curves

Component intersection curves are easily extracted automatically from the component tag information for each face. Going through the edge-face list, an edge is flagged if the two adjacent faces belong to different components. Such an edge must lie on the intersection between neighboring components. The set of flagged edges are concatenated automatically to form intersection curves.

High Curvature Contours

A completely automated scheme to detect high curvature contours such as those along leading edges of wings and nacelles is difficult since tests based on a single edge of a triangle alone are not sufficient. A scheme has to be designed to examine the total turning angle over a local surface region. Currently, a robust method to define such a local region is not available. Until such a scheme is developed, an alternative approach used in this paper is to utilize a graphical interface called OVERGRID[16] and a stencil-walk scheme. Inside OVERGRID, the user selects two vertices on a high curvature contour by visual inspection. A curve between the two vertices is automatically constructed by a stencil walk from one vertex to the other. The walk is constrained to lie on existing edges of the triangulation and utilizes the vertex-edge list to walk from one vertex to the next. In the edge selection process for the path, a higher preference is given to an edge that separates two faces with higher normal vector deviation than those from alternative edges. Curved leading edges, such as those found on nacelles, can thus be captured by this method (see Figure 3).

Open Boundary Curves

Open boundary curves are easily extracted from the edge-face list by automatically concatenating edges with just one adjacent face.

Corner Points

The seam curves identified by the above four methods are all extracted from existing edges of the surface triangulation. Identification of point coincidence can thus be performed in index space. A seam curve is split at an interior point if such a point coincides with the end point of another seam curve. After the split, a point where three or more end points of seam curves coincide is flagged and extracted. The degree of a corner point is defined to be the number of seam curves connected to the point.

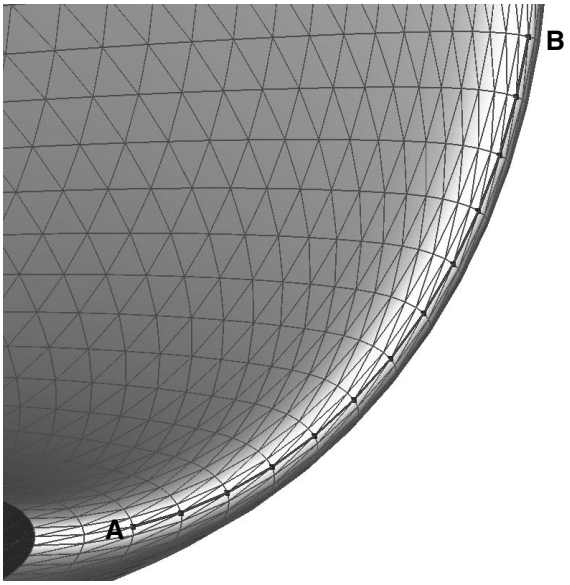


Fig. 3: Surface curve created along nacelle leading edge by connecting existing edges between vertices A and B.

Apex Points

An apex point on a surface is topologically equivalent to the apex of a cone. Since the surface is locally smooth other than at the apex point, there are no seam curves connected to the point. In order to determine if a vertex on a triangulation is an apex point, a vertex normal is first computed by averaging the surrounding face normals using the vertex-face list. The point is an apex point if the angles between vertex normal and all connected edges are larger than some threshold (taken to be 90 degrees in this paper).

2.3. Other Surface Curve Creation Schemes

For reasons explained below, it is convenient to construct two other methods for creating surface curves on triangulations. These are described in the following two subsections. With the surface curve creation schemes presented in Section 2.2, points on the resulting curves are all derived from vertices on the triangulation. This allows for storage of these curves in terms of the index space of the triangulation vertices. For the surface curve creation schemes described in this section, points on the curves can be arbitrarily located and do not admit index space storage.

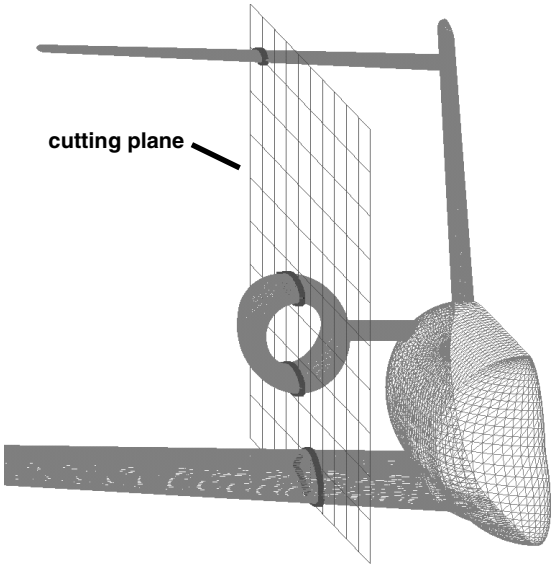


Fig. 4: Surface curves created on a business jet geometry by intersection with a single Cartesian plane.

Sectional-Plane Cut

For geometries with no surface features (e.g., a completely smooth closed surface), or where no smooth seam curves are available (e.g., a poorly resolved surface with many faceted faces or many minute surface features), surface curves are more easily created by intersection with a cutting plane. Currently, only intersection with a Cartesian x, y, or z cutting plane has been implemented. The set of faces that intersects the specified Cartesian cutting-plane can be easily extracted by examining the bounding box of each face. Each such face may intersect the cutting plane at 2 distinct points (2 edge intersections), along an entire edge (edge is coplanar with cutting plane), at 1 distinct point (vertex intersection), or along the entire face (face is coplanar with cutting plane). The last two cases do not produce line segments and are disregarded. The remaining two cases produce a line segment for each face/cutting-plane intersection. These line segments are automatically concatenated to create one or more sectional curves for a single cut. Figure 4 shows four curves created by a single Cartesian plane intersection through a business jet geometry.

Direct Path Between Two Vertices

After surface grids are created from the seam curves, gaps are sometimes left uncovered in the smooth regions of the surface. An algorithm to automatically cover these gaps for a multiple panel network surface description is given in Ref. 9. Unfortunately, the scheme cannot be easily extended to cover gaps on a triangulated surface description. The remedy in this case is to construct surface curves along one or more boundaries of the gaps and then cover the gaps using hyperbolic or algebraic schemes.

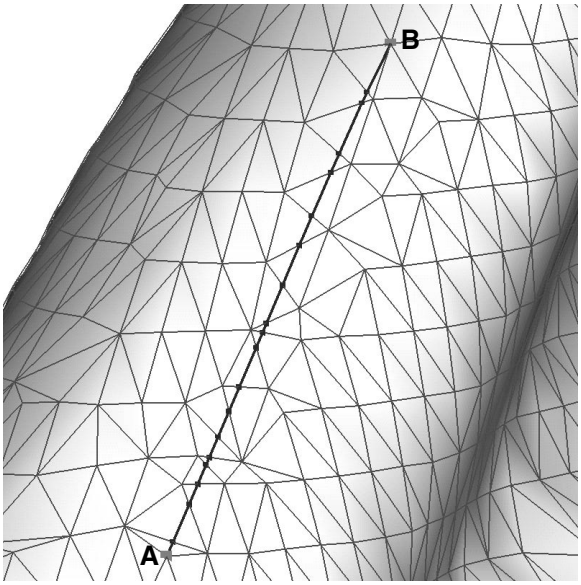


Fig. 5: Surface curve created by connecting line segments along direct path between A and B.

An arbitrary surface curve can be constructed between two existing vertices, A and B, on a surface triangulation. Starting at vertex A, the vertex-face list is used to find the face F_1 that contains the most direct path to vertex B. The resulting path through F_1 either (1) follows an edge of F_1 or (2) intersects the edge of F_1 that is opposite to vertex A. In case (1), the end point of the path in F_1 is another vertex and the same procedure can be followed to find the next segment of the path. In case (2), the end point of the path in F_1 is a point on an edge. The edge-face list is used to locate the neighboring face F_2. The most direct path to vertex B through F_2 is then computed. Again the end point of this path through F_2 could land on another edge or at a vertex. The above procedure can then be repeated until vertex B is reached (see Figure 5).

2.4. Surface Domain Decomposition

In this paper, the automatic domain decomposition scheme described in Ref. 9 is extended. Previously, only the corner points are identified and a polar grid topology is placed around all such points. In this paper, apex points as described above are also identified. It is proposed that any singular point of a surface can be classified as a corner point or an apex point. Furthermore, the surface region around such singular points can be covered using a surface grid with an O-type or H-type topology. The O-type topology is identical to those found in the polar grids described in Ref. 9. This topology can be used on any singular point and is the only option for corner points of degree 4 or higher, and at apex points. For a corner point of degree 3 or 4, either the O-type or H-type topology may be applicable as shown in Figure 6.

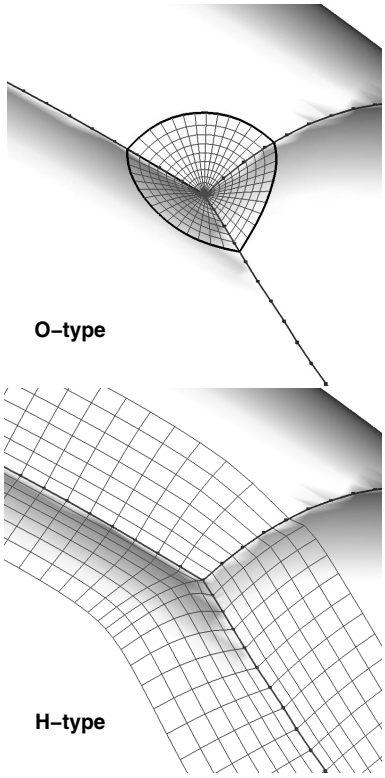


Fig. 6: O- and H-type grid topologies at a corner point. The three seam curves connected to the corner point are also shown.

The H-type topology is preferable when the seam curves are approximately orthogonal at a corner. Some heuristic rules are given below to automatically determine the choice of topology for a corner of degree 3. Similar criteria can be established for corners of degree 4.

(1) When two of the three seam curves lie along an open boundary of the geometry, there are only two angles separating the three seam curves. An H-type topology is selected if the ratio of the two angles is less than R where R is chosen to be 1.3 for this paper.

(2) When the corner is in the geometry interior, there are three angles a, b, c separating the three seam curves. Let R_{ab} be the ratio between angles a and b and let similar definitions holds for R_{bc} and R_{ca}. An H-type topology is selected if (a) the ratio of one angle with π is less than R and the ratio between the remaining two angles is also less than R, or (b) R_{ab}, R_{bc} and R_{ca} are all less than R.

After the seam points are classified, the procedure proceeds as described in Ref. 9 to retract seam curves from the O-type singular points, and to redistribute grid points on the retracted seam curves. At H-type corners, two of the seam curves have to be reconnected (see Section 3.2). The result is a decomposition of the surface domain around the surface features described in Section 2.2.

2.5. Grid Point Redistribution

An automatic scheme is used to redistribute grid points on all surface curves such that the following criteria are satisfied.

(1) The maximum grid spacing is no larger than a given value.

(2) Grid points on the redistributed curve lie on the piece-wise linear segments of the original curve; and grid points on the original curve are no more than a specified distance from the piece-wise linear segments of the redistributed curve. This ensures that grid spacings are automatically reduced at high curvature regions.

(3) Sharp turns on the original curve are preserved.

(4) Local grid spacings are reduced at sharp turns.

(5) Grid spacings vary smoothly along the redistributed curve.

3. Grid Generation Scheme

A marching scheme is used to create a surface grid from an initial curve or an initial point. For surface curves not connected to an H-type corner, a basic hyperbolic method[2,3] can be used (see Section 3.1). For surface curves connected to an H-type corner, the basic hyperbolic method is still employed but a corrector smoothing scheme is added (see Section 3.2). For an initial point (O-type corners), a special algebraic scheme is used to generate the first grid layer away from the point, and a hyperbolic method is used on the subsequent layers (see Section 3.3).

3.1. Marching Scheme from Surface Curves

In hyperbolic grid generation, a grid is created by marching from an initial curve over a specified distance via a sequence of grid spacings. The sequence of grid spacings is conveniently prescribed using a hyperbolic tangent stretching function which provides control of grid spacings at the initial curve and at the outer boundary. At each step, starting from the initial curve, the marching scheme produces a new layer of points. This new layer of points is then projected onto the reference surface. The marching and projection procedures are repeated until the final distance is reached.

In Ref. 2, a projection algorithm was described for a reference surface consisting of multiple panel networks. In this paper, the same marching scheme is used but an algorithm is presented below for projection onto a triangulated surface. Prior to generating any grids, a pre-processing step is performed once to create a set of face bounding boxes and a layer of prisms over the faces of the surface triangulation.

Each face bounding box is simply the Cartesian-axes-aligned bounding box for the face with a tolerance

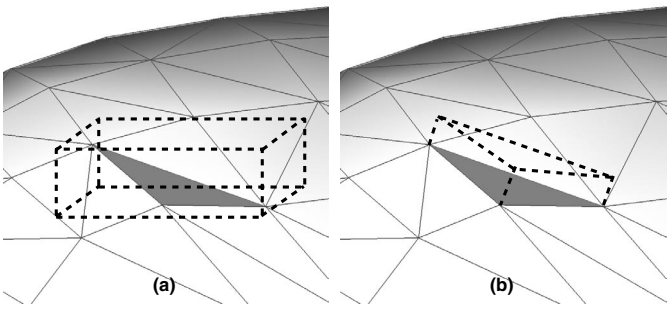


Fig. 7: Entities used by the projection algorithm for a surface triangle. (a) Extended bounding box. (b) Bilinear prism.

added to all 3 directions (see Figure 7a). This ensures a finite volume for the face bounding box even if the face is coplanar with a Cartesian plane. Furthermore, any arbitrary point close to the surface should be contained inside at least one of the face bounding boxes (see below for a mechanism that guarantees this condition).

The prisms are formed by extruding each vertex by a distance D along the vertex normal which is computed by averaging the normals of the surrounding faces (see Figure 7b). As will be shown later, the exact value of D is not important as long as it is small relative to the size of the entire geometry. For this paper, D is chosen to be 0.001 times the geometry's bounding box diagonal.

Let a point P be created by the marching scheme during hyperbolic grid generation. The projection procedure onto the surface is divided into several steps.

(1) A list of candidate faces whose bounding boxes contain P is determined. If P is found to be outside of all face bounding boxes, each face bounding box is automatically expanded in all 3 directions by a small fraction of its original dimensions. The process is repeated until at least one face bounding box is found for P. Note that the face bounding box expansion is only needed when P is not close to the surface. Such situations will only occur if the marching step is large relative to the local surface curvature. In other words, the grid spacing is too large for accurate surface resolution - a situation that should be avoided. Hence, a warning is issued to the user if the automatic bounding box expansion extends beyond several levels.

(2) For each face in the candidate list, a Newton iteration is performed to compute interpolation coefficients in a bilinear representation of the prism[17] over the face. The interpolation coefficient in the normal direction is disregarded in testing for convergence of the iteration. In other words, the exact height D of each prism is irrelevant in determining whether the point belongs to the prism. This procedure generates a list of prisms where each prism contains the point.

(3) For most cases, only one prism will be found in the list from step 2. The interpolation coefficients in the tangential direction are then used to project the point to the surface triangle. If multiple prisms are found, the one that gives the shortest distance between P and the surface triangle is chosen.

The projection scheme described above is found to be robust and reasonably fast. Numerical experiments show that it took 3 CPU seconds on an SGI R10000 175 MHz workstation to generate a surface grid with 2436 points on a triangulation containing 4005 vertices and 8006 faces. Another surface grid with 950 points was created in 6.5 CPU seconds on a triangulation containing 37700 vertices and 75000 faces.

3.2. Marching Scheme at H-type Corners

In this section, a scheme for treating an H-type corner of degree 3 is described. Simple extensions can be made to handle H-type corners of degree 4. After the heuristic rules given in Section 2.4 are used to determine that a corner is of H-type, two of the seam curves connected to the corner are reconnected automatically to form an initial curve for hyperbolic marching. The remaining curve becomes a reference curve that an interior grid line of the hyperbolic grid must follow during the marching procedure. In Figure 8a, curves C1, C2 and C3 are reconnected to form an initial curve running through H-type corners at H1 and H2. Curves C4 and C5 become interior reference curves for the resulting initial curve.

Given an initial curve, occurrence of an H-type corner in an interior point J_c is automatically detected by searching for end points of reference curves coinciding with J_c. After each hyperbolic marching and projection step, the predicted point along the grid line emanating from J_c is projected onto the reference curve. Several smoothing steps are performed locally and the smoothed points are projected back onto the reference surface. Figure 8b shows the surface grid produced from an initial curve with two interior H-type corners.

3.3. Marching Scheme at O-type Corners

For an O-type corner or an apex point, a polar topology is used for the surface grid wrapping around the initial point. This subsection outlines a special algebraic scheme to generate the first layer of points away from initial point. The hyperbolic marching method described in Section 3.1 is employed for the subsequent layers. A mechanism similar to that presented in Section 3.2 is used to guarantee that each seam curve

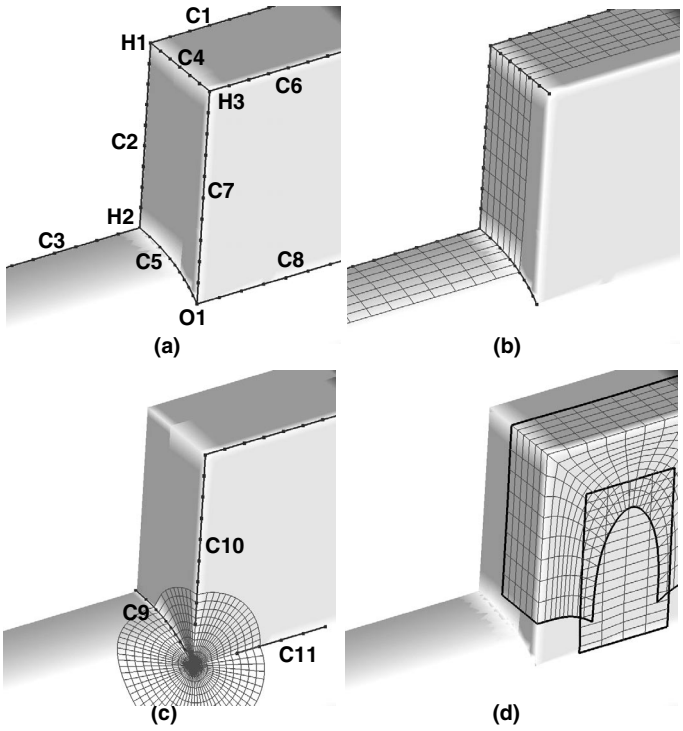


Fig. 8: Curves and surface grids at H-type and O-type corners. Curves are labelled Ci with with $i = 1, 11$. Corners are labelled Hi with $i = 1, 3$ and O1. (a) Initial and reference curves. (b) Surface grid from H-type corners. (c) Surface grid from O-type corner. (d) Surface grids from nearby retracted seam curves.

connected to the initial point is followed by a radial surface grid line during hyperbolic marching.

Since all seam points are assumed to coincide with a vertex on the surface triangulation, the vertex-edge list is used to find all edges connected to the initial point. Then the ordering of the edges around the initial point and the angle between each pair of adjacent edges are determined. The circumferential direction is then divided into sectors where each sector is separated from the next by a seam curve connected to the initial point. For an apex point, there is just one sector. The angle subtended by each sector is then computed by summing the angles between the corresponding ordered edges connected to the initial point.

Next, the number of points in the circumferential direction in each sector is automatically determined. Based on the polar topology of the grid, the arc length along the outer boundary of each sector can be estimated from the marching distance and the angle subtended by the sector. The number of points in the circumferential direction in the sector is computed by assuming a uniform spacing along the outer boundary no larger than the end grid spacing in the marching direction.[9]

Grid points on the first layer are created by extruding radially on the surface from the initial point to a distance equal to the prescribed initial spacing for this surface grid. The angular positions of points in the first layer in each sector are made to form a uniform distribution over the angle subtended by the sector.

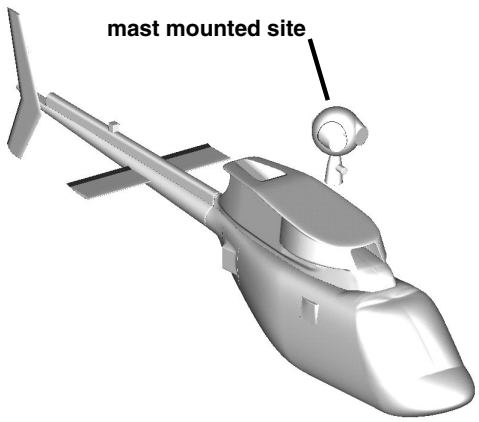


Fig. 9: Rotorcraft surface geometry.

In Figure 8c, radial grid lines of the surface grid grown from O1 are made to follow C5, C7 and C8. Curves C9, C10 and C11 are the retracted versions of the connected seam curves. Other surface grids are grown from these curves as shown in Figure 8d.

4. Test Cases

Three test cases are presented below to illustruate the schemes described in the previous sections. These include a rotorcraft, a space vehicle and a triceratops.

4.1. Rotorcraft

The first test case is a rotorcraft shown in Figure 9. For a complex configuration such as this, it is usually easier to treat the problem in several parts. A fine surface triangulation for the mast mounted site containing 61000 faces and 30000 vertices is shown in Figure 10a.

Seam curves are automatically extracted around the surface discontinuities and intersection curves of the attachments. Several other surface curves are extracted by sectional cuts to create more initial curves for hyperbolic marching so that the surface can be covered completely (see Figure 10b). The hyperbolic grids shown in Figure 11 are generated via the OVERGRID graphical interface in less than 3 hours of manual effort. A total of 17 surface grids are created with 28660 points. About two thirds of the time is used in adjusting marching distances so that the surface is completely covered with proper overlap between neighboring grids. Similarly, 23 surface grids for the fuselage are created in about 5 hours of wall clock time (see Figure 12). The fuselage reference triangulation contains 226000 faces and 114000 vertices while the total number of points in the resulting surface grids is about 18000.

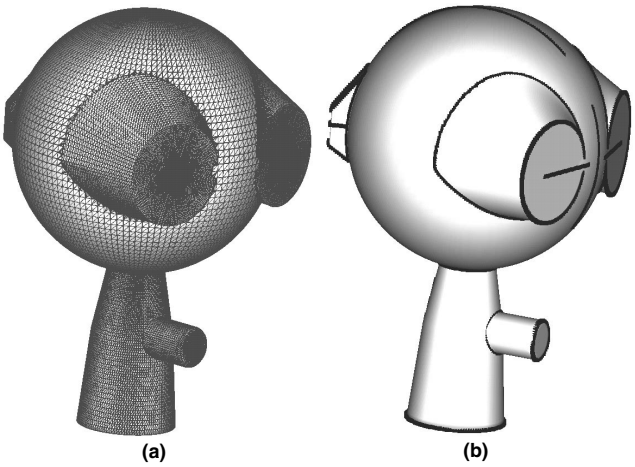


Fig. 10: Rotorcraft mast mounted site. (a) Surface geometry. (b) Seam and sectional cut curves.

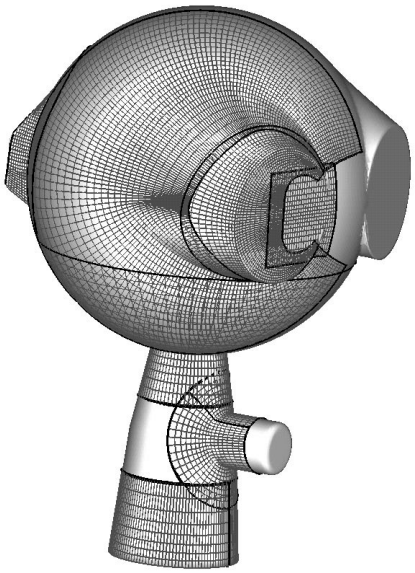


Fig. 11: Hyperbolic surface grids for rotorcraft mast mounted site. Not all grids are shown to improve display quality. Grid boundaries are depicted by dark lines.

4.2. Space Vehicle

The second test case is an experimental space vehicle shown in Figure 13. Only the left half of the symmetric vehicle is considered here. The surface geometry is highly resolved with about 75000 faces and 38000 vertices. Both fins have finite thickness trailing edges. Seam curves are automatically extracted around all sharp surface edges. The stencil-walk scheme is needed to create the surface curves over the high curvature contours along the leading edges of the fins, and the lower side walls of the body flap (see Figure 14).

Grids around most of the sharp surface features of the space vehicle, other than ones on the outboard fin, are shown in Figure 15. Generation of the 16 grids displayed took about 4 hours using the OVERGRID interface. The difficulty with this geometry lies in the moderately complex arrangement of seam curves and

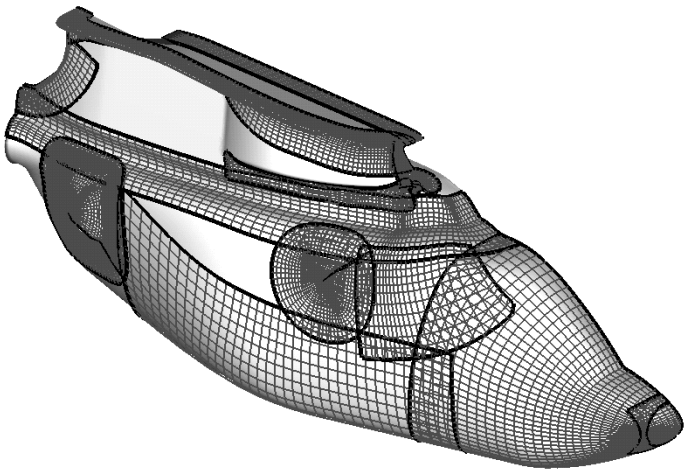


Fig. 12: Hyperbolic surface grids for rotorcraft fuselage. Not all grids are shown to improve display quality. Grid boundaries are depicted by dark lines.

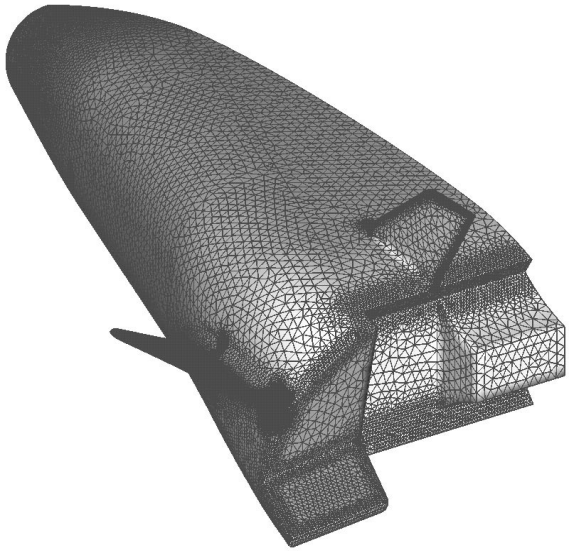


Fig. 13: Space vehicle geometry with about 75000 faces and 38000 vertices.

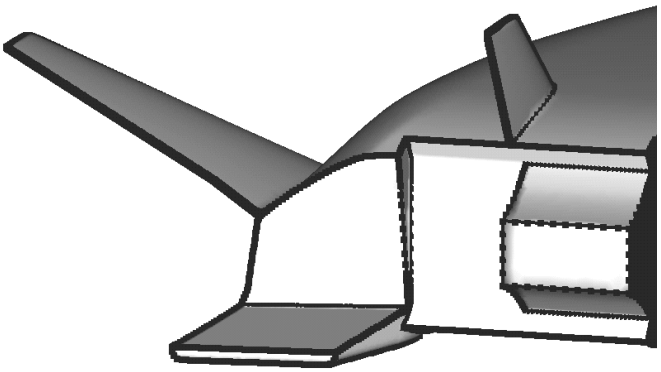


Fig. 14: Seam curves for the space vehicle viewed from the back.

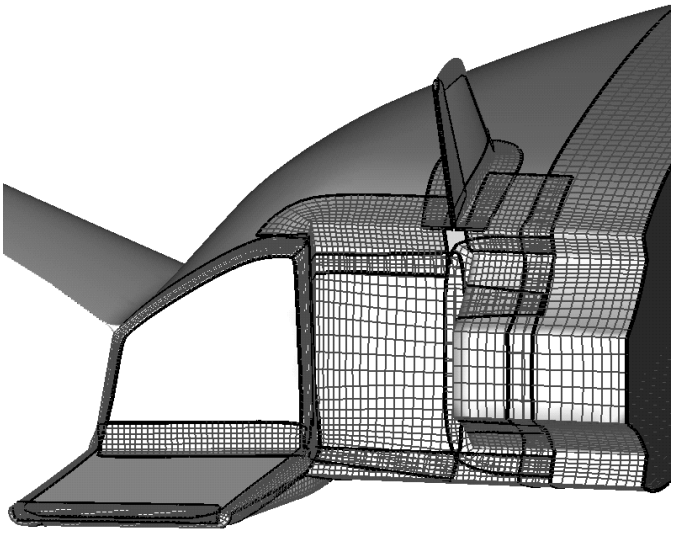


Fig. 15: Surface grids around seam curves in the base region of the space vehicle (grid boundaries depicted by dark lines).

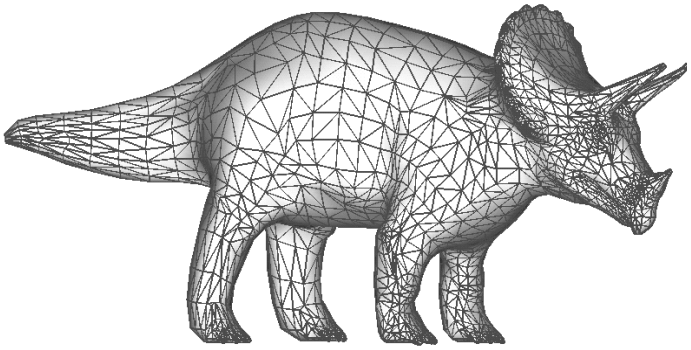


Fig. 16: Triceratops surface geometry with 5660 faces and 2832 vertices.

seam points in the base region that need to be captured. Also, time is needed to create surface curves along the high curvature contours in several places. It is anticipated that creating grids for the outboard fin and filling the remaining smooth parts of the surface geometry would require approximately another 4 hours of user time.

4.3. Triceratops

In the previous two test cases, the surface geometry is essentially smooth in most places except for certain local regions, and accurate representation of local discontinuities and high curvature regions are of high importance. For third test case, a mostly faceted irregular surface is chosen to illustrate the capability of the current scheme applied to a different class of geometry. A triangulation over a triceratops (Figure 16) is chosen for this purpose. Here it is not critical that all facets of the surface be captured accurately. The surface geometry contains 5660 faces and 2832 vertices.

Since the surface consists of many facets, initial

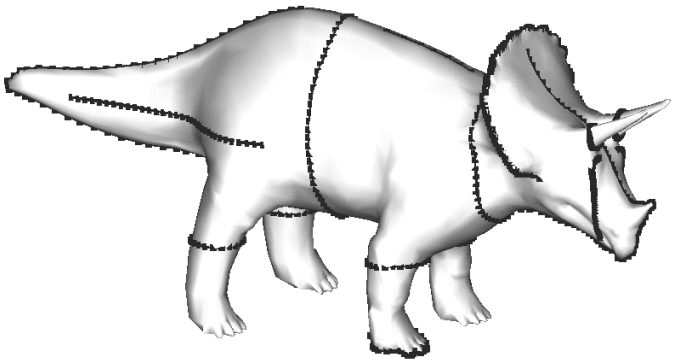


Fig. 17: Triceratops surface curves.

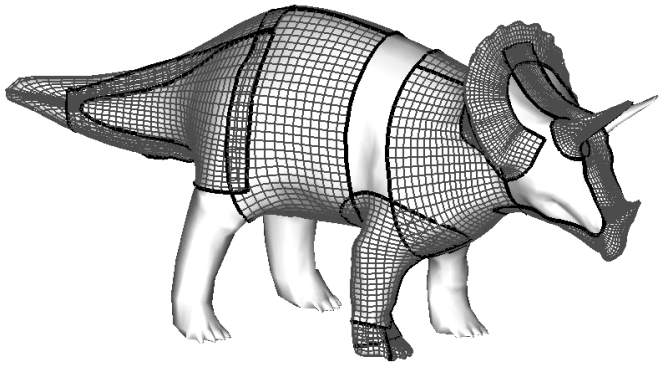


Fig. 18: Surface grids for triceratops. Not all grids are shown to improve display quality. Grid boundaries are depicted by dark lines.

curves for the hyperbolic marching scheme are created by various sectional-plane cuts on the body, and by the stencil-walk method along the crown of the head and the base of the feet (Figure 17).

All surface grids are obtained by hyperbolic marching, a sample of which are shown in Figure 18. A special option in the grid generator is used to close the outer boundary of the marching scheme to a point for grids around the two sharp horns and four feet. The complete configuration consists of 17 surface grids and are created with about 14000 grid points. Starting from the surface triangulation, the process of surface curve creation and surface grid generation took about one and a half hours of user time using the OVERGRID graphical interface.

5. Concluding Remarks

A scheme is presented for generating structured grids over a triangulated surface geometry definition. Surface features are classified into seam curves and seam points and are captured under an overset domain decomposition procedure. Various methods for creating surface curves for triangulated surfaces are also described. A hyperbolic marching scheme is used to generate grids from initial curves and points. Special treatments are needed at the initial points which can be of O- or H-type. A robust and efficient surface

projection scheme that is coupled to the hyperbolic grid generator is also presented. It is shown that grids for moderately complex configurations can be generated in a reasonable amount of time. The human effort needed to model CAD geometry data with computational grids, and thus the design cycle time, is reduced with this new capability.

A surface triangulation is a popular geometry representation that can be easily derived from CAD models as well as scanned objects. While this opens up the potential for tackling a wide range of problems, the lack of a standard format can be annoying to the user. In addition to the CART3D and FAST formats mentioned in this paper, other commonly used formats include stereolithography (STL), finite element neutral (FNF), and Virtual Reality Markup Language (VRML). Varying amounts of information are present in the different formats while some also allow duplicated vertices. Degeneracies such as self-intersections occur quite frequently and require pre-processing work to remove. In order to truely streamline the geometry handling and grid generation process, the user community has to agree on a robust standard format for surface triangulations.

Acknowledgements

The author is grateful to Michael J. Aftosmis and Dr. Shishir Pandya for many valuable ideas and discussions on surface triangulations. This work was performed when the author was employed by MCAT, Inc. under NASA contract number NAS2-14109, Task 23. Funding for the work was provided by the DOD HPCMP CHSSI CFD-4 program.

References

1. Steger, J. L., Dougherty, F. C. and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, K. N. Ghia and U. Ghia, eds., ASME FED-Vol. 5, June, 1983.

2. Chan, W. M. and Buning, P. G., "Surface Grid Generation Methods for Overset Grids," *Computers and Fluids*, **24**, No. 5, 509–522, 1995.

3. Chan, W. M., "Hyperbolic Methods for Surface and Field Grid Generation," *CRC Handbook of Grid Generation*, Eds. Thompson, Soni and Weatherill, CRC Press, 1998.

4. Chawner, J. R. and Steinbrenner, J. P., "Automatic Structured Grid Generation Using GRIDGEN (Some Restrictions Apply)," *Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions*, NASA Lewis Research Center, NASA CP-3291, 463–476, 1995.

5. Wulf, A. and Akdag, V., "Tuned Grid Generation with ICEMCFD," *Surface Modeling, Grid Generation, and Related Issues in Computational Fluid Dynamic (CFD) Solutions*, NASA Lewis Research Center, NASA CP-3291, 477–488, 1995.

6. Haimes, R. and Follen, G., "Computational Analysis PRogramming Interface," *Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations*, Eds. Cross, Eiseman, Hauser, Soni and Thompson, July, 1998.

7. Aftosmis, M. J., Berger, M. J. and Melton, J. E., "Adaptive Cartesian Mesh Generation," *CRC Handbook of Grid Generation*, Eds. Thompson, Soni and Weatherill, CRC Press, 1998.

8. Aftosmis, M. J., Delanaye, M., and Haimes, R., "Automatic Generation of CFD-Ready Surface Triangulations from CAD Geometry," AIAA Paper 99-0776, 1999.

9. Chan, W. M. and Gomez, R. J., "Advances in Automatic Overset Grid Generation Around Surface Discontinuities," AIAA Paper 99-3303, *Proceedings of the AIAA 14th Computational Fluid Dynamics Conference*, Norfolk, Virginia, 1999.

10. Meakin, R. L., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," AIAA Paper 93-3350, *Proceedings of the 11th AIAA Computational Fluid Dynamics Conference*, Orlando, Florida, 1993.

11. Atwood, C. A. and Van Dalsem, W. R., "Flowfield Simulation About the Stratospheric Observatory For Infrared Astronomy," *J. of Aircraft*, **30**, No. 5, 719–727, 1993.

12. Duque, E. P. N. and Dimanlig, A. C. B., "Navier-Stokes Simulation of the RH-66 Comanche Helicopter," *Proceedings of the 1994 American Helicopter Society Aeromechanics Specialist Meeting,* San Francisco, California, 1994.

13. Slotnick, J. P., Kandula, M. and Buning, P. G., "Navier-Stokes Simulation of the Space Shuttle Launch Vehicle Flight Transonic Flowfield Using a Large Scale Chimera Grid System," AIAA Paper 94-1860, *Proceedings of the 12th AIAA Applied Aerodynamics Conference*, Colorado Springs, Colorado, 1994.

14. Rogers, S. E., Cao, H. V. and Su, T. Y., "Grid Generation for Complex High-Lift Configurations," AIAA Paper 98-3011, June, 1998.

15. Walatka, P. P., Clucas, J., McCabe, R. K., Plessel, T. and Potter, R, "FAST User Guide," RND-93-010, NASA Ames Research Center, 1994.

16. Chan, W. M., "OVERGRID – A Unified Overset Grid Generation Graphical Interface," submitted to *Journal of Grid Generation*, 1999.

17. Dhatt, G. and Touzot, G., *The Finite Element Method Displayed*, Wiley, 1984.