IN-61

143110

P- 15

VGRIDSG
An Unstructured Surface Grid Generation Program

Michael J. Bockelie

Computer Sciences Corporation
Hampton, VA

Prepared For
Langley Research Center
under Contract NAS1-19038

January 1993

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

VGRIDSG - An Unstructured Surface Grid Generation Program *

M..J.Bockelie [†]

Abstract

This report contains an overview of the VGRIDSG unstructured surface grid generation program. The VGRIDSG program was created from the VGRID3D unstructured grid generation program developed by Vigyan, Inc.. The purpose of this report is to document the changes from the original VGRID3D program and to describe the capabilities of the new program.

I. Introduction

The VGRIDSG program was developed at the request of Dr. Robert E. Smith (ACD/CAB). The primary objective of this project was to modify the VGRID3D code to utilize a surface definition based on a structured set of bi-cubic Hermite patches as used in the GRIDGEN structured grid generation software system. Some secondary objectives of the project were: to incorporate the ability to read in surface definitions in standard point and CAD data files as used within the Numerical Geometry Laboratory (GEOLAB) at NASA Langley; to develop the ability to compute the surface grid within a 2D parametric space representations of the surface as used in GRIDGEN; and to incorporate additional graphics routines as needed to assist the user in generating the surface grid.

The original objectives of this project have been achieved. A beta version of the VGRIDSG is available within GEOLAB and at AAD/TAB. By design, the new code performs only surface grid generation. The grid computed by VGRIDSG can be input to the original VGRID3D program to generate a volume grid.

The new code should be viewed as a research code. The program has been extensively tested with debugging tools to check for coding (FORTRAN) errors, but has been used for only a few configurations. Further testing and evaluation is in order before attempting to release the code for production use.

The main body of this report contains, in order, a description of the original code, a description of the modifications, an example of using VGRIDSG to generate the surface grid for a High Speed Civil Transport (HSCT), and a discussion of suggested improvements.

II. Original Code

The baseline version of VGRID3D that was modified corresponds to the export graphics code (version 1.1, Oct. 1991) that executes on an SGI IRIS graphics workstation. The original code performs both surface and volume grid generation. The grid is generated using

---

an advancing front method in which the spacing parameters controlling the grid generation are interpolated from a coarse unstructured background grid. The vehicle surface is defined by a set of triangular and quadrilateral surface patches. For each patch, a surface is defined by a bi-linear blending of the curves that form the edges of the patch. Each edge of the patch is formed from a single curve. The curves are typically defined as simple piecewise continuous cubic splines. In addition, the curves are not allowed to close (i.e., the start and end points of the curve must be distinct). The surface grid is generated by first placing grid points on the curves that form the patch edges. After discretizing all of the edges, the interior of the patches are filled with triangles.

The result of using the simple bi-linear surface patches is that the vehicle surface often times has a rough appearance. In addition, the surface grid points will not lie on the actual surface of the vehicle. Not allowing closed contours to define a patch edge makes it impossible to accurately model a closed surface (e.g., a cylinder). Requiring each edge of a patch to be a single curve results in an unflexible patching scheme because it requires that patches abut only at the patch corners.

III. New Code

The new code (VGRIDSG) represents a substantial change from the original code. Altogether, about 18,000 lines of code have been added to the program. A multitude of changes have been made. The most notable change is the use of a surface definition based on networks of structured bi-cubic Hermite patches - the same approach used in the GRIDGEN software system. Using the networks of bi-cubic patches results in: a very smooth surface; a surface that more accurately models the true vehicle surface; and a surface definition based on "state of the practice" techniques used in aerospace CFD applications. Using this approach also allows for defining closed surfaces (e.g., cylinders, wings). Using the networks of bi-cubic patches also results in the code requiring substantially more memory and cpu time (factor of 5) to generate the surface grid. For further details on the mathematical formulations for the patch definitions, surface modeling, and surface grid generation see references [3-7].

The networks defining the surface are read from files containing the surface data in either point (VGRID,LAWGS,PLOT3D,GRIDGEN) or CAD (PATRAN neutral file) format. The tangential derivatives in the bi-cubic patch definition can be computed from a simple central difference or by passing a spline through the given surface points and then differentiating the spline. The splines can be computed using a standard technique (continuous curvature) or a monotone method (Akima). The cross (second) derivatives are computed as the average of the cross derivatives in each direction. The benefit of using this type of surface definition is that the surface will be C2 continuous within a network. Note that at network boundaries only C0 continuity can be assumed.

In the new code, the portions of the vehicle surface to be gridded are defined by a set of subregions. Each subregion is defined as a closed contour in uniform parameter space (i.e., index space). The closed contours can be of any shape and can be formed from multiple curves (connectors). The list of connectors defining the closed contour must be ordered such that the region to be gridded lies to the left when traveling along the contour - this same rule was required in the original code and is standard in advancing front methods. In addition, we require that all of the connectors forming a subregion be defined on the same network. As with the original code, the grid points are first placed on all of the connectors that define the subregions, after which the interior of the subregions are triangulated. For the case of gridding a closed surface or a surface containing a polar singularity in the data base (e.g., in a nose region), a point in physical space can have multiple representations in uniform parameter space. To account for this situation, we use a "ghost point" for any second representation of the physical point.

2

Neighboring subregions are only required to match at the end points of connectors (not corner points of a patch). This results in a very flexible surface patching scheme. However, special care must be used in defining the connectors that form the mutual sides of neighboring subregions. Because we require that all connectors used within a subregion be defined on the same network, the connectors that form the mutual side must be defined twice - once on each network. In the grid generation process, when placing points on the connectors the first instance of the mutual connectors is discretized using "true" grid points and the opposing connector is discretized using the ghost points. The special cases of a curve lying on degenerate edge of a network or a network edge corresponding to the edge of a closed surface are treated in a similar manner. At the end of the surface grid generation process the ghost points are eliminated so that the final surface grid contains a list of unique points in physical space.

The generation of the triangles in the interior of a subregion is performed in a 2D parameter space representation of the network. The grid can be generated in either a uniform, normalized arc length, total arc length, centripetal, Foley parameter space or a parameter space defined from a planar projection of the surface. In addition, the parameter space can be computed from the complete network or from only a subset of the network computed from the smallest box enclosing the subregion in uniform parameter space. Note that the parameter space used to generate the grid can have a dramatic effect on the quality of the grid, but has no effect on the surface definition. That is, regardless of the parameter space used to generate the grid, the final grid points lie on the original surface defined by the bi-cubic patches.

The following is an outline of the modifications made to the original VGRID3D code.

Surface Definition

1. all surfaces are defined by a network of structured patches.

2. the patches can be defined as bi-linear or bi-cubic Hermite:

   - given point data → computes patch representation;
   - given CAD data → use given patch representation.

3. a bi-cubic patch can be defined with a tensor product of splines or a simple central difference rule to compute derivative terms:

   - splines can be computed with standard method (continuous curvature), standard method plus a chain rule (see [5]), or monotone method (Akima);
   - multiple options for boundary conditions on spline definitions.

4. we do not require that all surfaces be computed in the same manner.

5. option to turn "off" un-needed networks in the surface definition.

Data Input

1. can use point or CAD data files to define surface:

   - acceptable point data formats: VGRID,LAWGS,PLOT3D, GRIDGEN;
   - acceptable CAD data formats: PATRAN neutral file.
   - can use multiple files to store data.

2. lines (connectors) that form edges of subregions to be gridded are input via uniform parameter space (e.g., i,j) representation rather than real space representation.

   - typically need only network number and start/stop ij indices to define a line.

3. many user input options for surface and subregion definitions

   - allows adjusting the grid generation process in difficult problems.

4. contains rigorous checking of input data and outputs informative messages when an error occurs.

5. use routine by B.Wedan (VIGYAN, Inc.) to read .d3m file that allows flexibility of placing additional comment statements directly into the input file.

6. a "quick reference sheet" is available that describes the format and items in the new .d3m file.

Graphics and Visualization

1. options to plot data base networks, connectors, subregion edges, ghost points and 2D parameter space representation of surface subregion to be gridded.

   - curves can be drawn as simple linear point connect or as smooth curve employing user defined point resolution.

2. options to output data base network and back ground grid spacing and stretching parameters projected onto the data base points to PLOT3D binary, multizone, structured grid and function data files that can be read and displayed in FAST.

Data Output

1. options to output surface grid data in file formats compatible with following visualization tools:

   - DEMAC (fort.22, front.rst [alt.])
   - FAST (fort.fast)
   - SURFACE (fort.fast, fort.22)
   - TECPLOT (fort.tect)

2. option to output restart files used by VGRID3D (front.rst,coord.rst,file.int,file.bc)

3. output to a file (fort.23) the uniform parameter space (network number, u,v location) and real space (xyz) point definitions for the grid, point connectivity, as well as the line and subregion number on which the grid point is located.

4. ability to designate an integer flag for each subregion to simplify displaying of selected portions of the surface grid.

   - particularly useful when displaying flow solution data
   - based on method developed by C.R.Gumbert (AAD/CMB)

5. options to remove or leave the ghost points in the surface grid.

   - options to check that all grid points are uniquely defined in real space

6. output a file (fortsg.flag) containing boundary condition flags for use by USM3D pre-processing program PREFLO.

Definition Of Subregion To Be Gridded

1. subregion can be defined by an arbitrary number of lines (connectors):

   - the list of connectors must form a closed circuit in uniform parameter space.
   - connectors can have zero length (e.g., along a network edge containing a singularity).

2. subregion edges can meet in a "T".

   - edges not restricted to meeting only at patch corners.

3. note that in the new code the subregion to be gridded is assumed to lie on the vehicle surface and does not effect the surface definition.

   - in original VGRID3D, the surface patches performed the dual role of defining the vehicle surface and defining a portion of the surface to be gridded.

2D Parameter Space Representation

1. input options allow generating the surface grid in a 2D space defined as:

   - uniform parameter space (UPS, index space).
   - normalized arc length parameter space (NAPS, a [0,1]x[0,1] space).
   - total arc length parameter space (TAPS) .
   - centripetal parameter space (CPS, a [0,1]x[0,1] space).
   - Foley parameter space (FPS, a [0,1]x[0,1] space).
   - planar projection parameter space (PPPS).

2. options are available for computing the 2D space using the complete network on which the subregion exists, or to use a subset of the network.

   - subset computed as smallest rectangle containing the uniform parameter space representation of the subregion.

Node Spacing and Metric Conversion

1. The node spacing can be defined using either a structured or unstructured back ground grid.

   - the structured background grid uses the point and line source technique developed by S.Pirzadeh (VIGYAN, Inc.).
   - for structured background grid, have option to specify dimensions or compute finest Cartesian grid that will fit into the allocated storage.

2. several options for determining node spacing in interior of subregion:

  • simple metric conversion based on ratio of length of current face in 2D parameter space to length of current face in 3D real space.

  • conversion factors determined by accounting for "stretching" and change of angle due to mapping of surface from 3D space to 2D space.

    – options to use "stretching" computed in manner equivalent to original VGRID3D and with formulation that is "mathematically correct".

3. developed a telescoping point search algorithm that significantly reduces cpu time to transform from 2D parameter space to 3D real space.

  • to perform transformation, must map a point in the non-uniform 2D parameter space (in which the grid is generated) to the uniform parameter space in which the surface is defined.

  • this step is used extensively when utilizing the option for computing the node spacing that accounts for the "stretching".

4. developed error recovery routine to handle situations where the above searching scheme fails to correctly locate the uniform space representation of a point.

  • can occur for points that lie on (or very near) the edge of a network.

IV. Example of Using VGRIDSG

This section contains an example of using VGRIDSG to generate the surface grid for a Mach 3.0 HSCT. For this problem the vehicle surface is defined by three networks. The first and second network extend from, respectively, the nose to the crank and from the crank to the trailing edge of the wing. The third network defines the boattail added to the wing/body configuration in order to conduct wind tunnel tests. The surface grid consists of 12 surface "subregions". The vehicle surface is defined by six subregions and the (planar) far field boundaries and plane of symmetry are defined by the other six subregions (see Figure 1). The subregions defining the vehicle surface have been created by splitting the three networks into upper and lower regions. The ability to let the subregions meet in a "T" can be seen in Figure 1 for the edge that is common to the wing and boattail subregions. The point and line sources used in the structured background grid have been placed so as to provide finer resolution along the wing edges and along the wing/fuselage intersection region (see Figure 2). The computed surface grid is illustrated in Figures 3 and 4 for portions of the upper and lower vehicle surface. Illustrated in Figure 5 is the grid that lies in the plane of symmetry and is near the vehicle surface. The surface grid illustrated herein is a "moderate" resolution grid containing a total of 11806 triangles (8520 on the vehicle) and 5905 points (4715 on the vehicle). This grid requires about 11 cpu minutes to compute in serial mode on a Silicon Graphics IRIS 4D Workstaion containing a series 400 processor. Using this surface grid as the initial front for the VGRID3D volume grid generator would result in a volume grid containing about 200K tetrahedra.

6

V. Recommendations

The VGRDISG code requires further testing and improvement if it is to be used in a production mode. The following is a list of items that should be implemented into VGRIDSG or other production programs that are intended to improve on the VGRID software system.

1. develop an interactive graphics code(s) that could be used to define the data required by the grid generator:

   - tool needs patch editing and restart capabilities of the existing PREGR1 tool, but should use "state-of-the practice" graphical user interface and computational techniques (e.g., panels for user input)
   - tool should have ability to compute and display properties of vehicle surface (e.g., curvatures)
   - ideally, the patch/subregion definition, background grid set up, and surface grid generation should all be in one tool

2. develop algorithm to compute connectivity table of the networks and then use this information to use the networks as the default for the vehicle surface subregion (i.e., automatic patching).

3. simplify the method of placing points on the edges of subregions that abut.

   - current procedure of defining lines on neighboring subregions that lie on different networks is somewhat clumsy and possibly too difficult for novice users to employ
   - automate by using a projection technique

4. further improve grid quality by incorporating:

   - triangle edge swapping
   - additional points in the circle criteria that is used to select the best point to form a triangle
   - grid smoothing procedures that increase the degree of smoothing in regions of high surface curvature

5. reduce the cpu time and memory to generate the surface grid:

   - re-organize computationally intensive operations
   - investigate use of distributed/parallel computing methods
   - alter "stretching" and metric conversion algorithms to not need to transform between 2D and 3D space
   - eliminate unneeded arrays to reduce data storage

6. improve program error recovery in subroutines that generate triangles:

   - new code has much better error recovery than original, but needs improvement
   - typically, if an error occurs, code prints a message and stops
   - alter code to print error message and then continue to next surface subregion

7

VI. Acknowledgements

VII. References

1. Parikh, P., Pirzadeh, S., Lohner, R., "A Package For 3D Unstructured Grid Generation, Finite Element Flow Visualization And Flow Field Visualization", NASA CR 182090, September 1990.

2. Pirzadeh, S., "Structured Background Grids For Generation Of Unstructured Grids Aby Advancing Front Method", AIAA Paper 91-3233.

3. Farin, G., *Curves And Surfaces For Computer Aided Geometric Design*, Academic Press, Second Edition, 1990.

4. Steinbrenner, J.P., Chawner, J.R., Fouts, C.L., "The GRIDGEN 3D Multiple Block Grid Generation System", WRDC-TR-90-3022, July 1990.

5. Abolhassani, J.S., and Stewart, J.E., "Grid Generation In A Parameter Space", AIAA Paper 92-2717.

6. Weatherill, N.P., "Grid Generation", *Numerical Grid Generation*, von Karman Institute For Fluid Dynamics, Lecture Series 1990-06, June 11-15, 1990, Rhode Saint Genese, Belgium.

7. Peraire, J., Morgan, K., Peiro, J., "Unstructured Mesh Methods For CFD", *Numerical Grid Generation*, von Karman Institute For Fluid Dynamics, Lecture Series 1990-06, June 11-15, 1990, Rhode Saint Genese, Belgium.
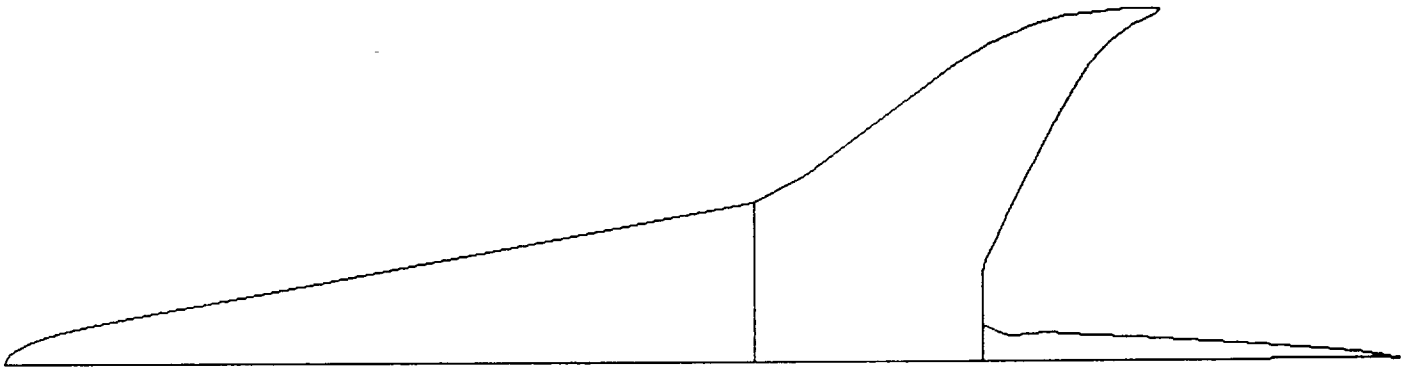
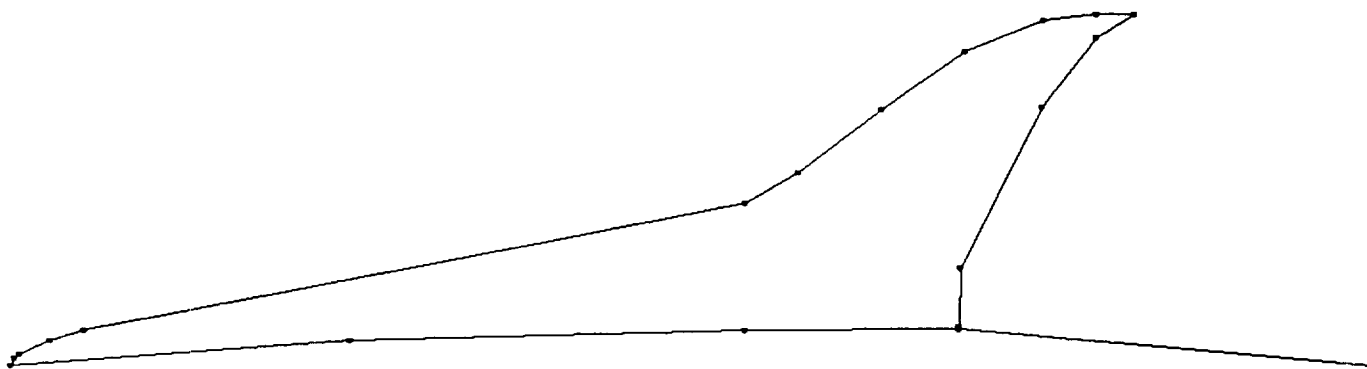FIGURE 1. Subregion boundaries for upper surface of HSCT.

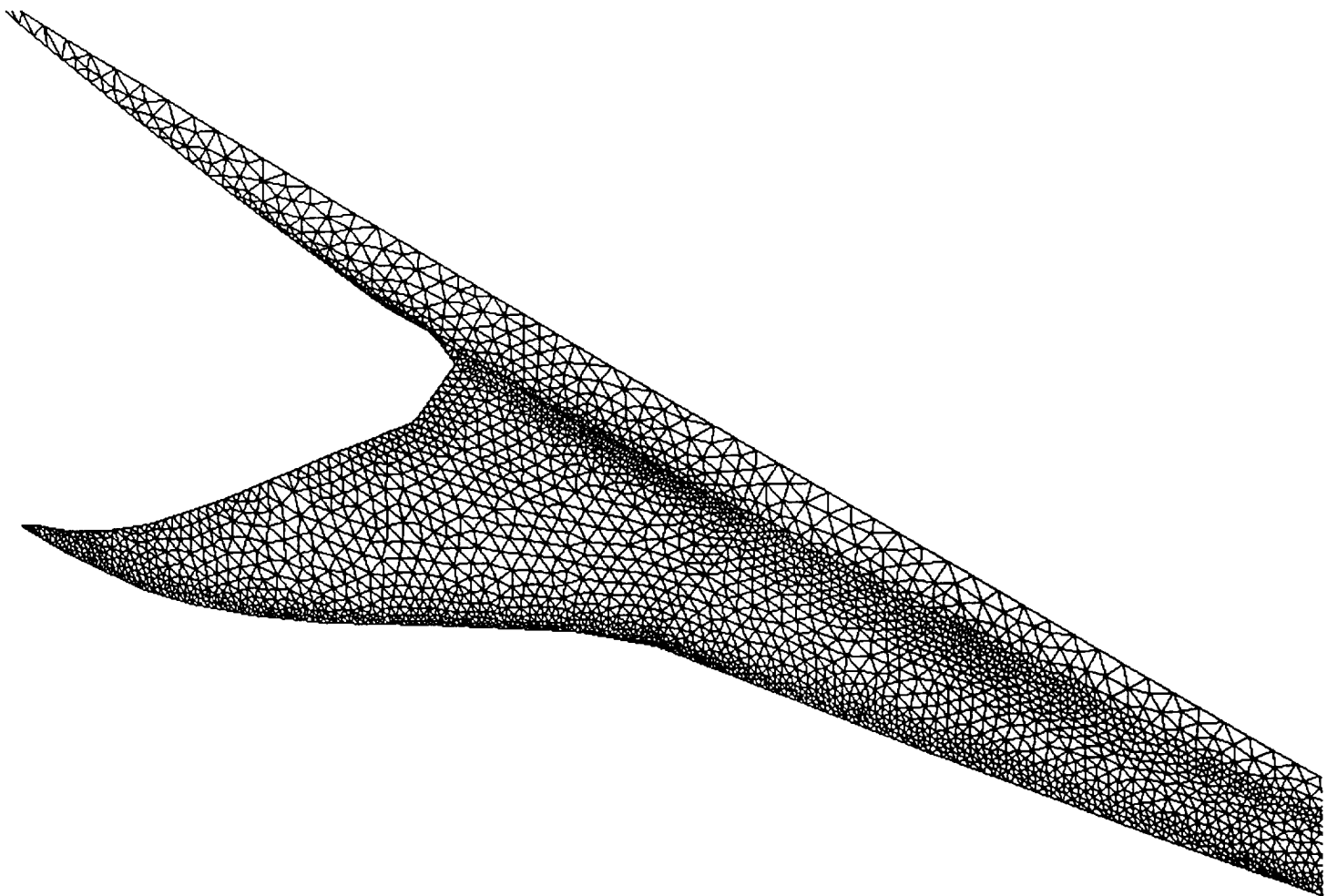FIGURE 2. Point and line source locations near vehicle.
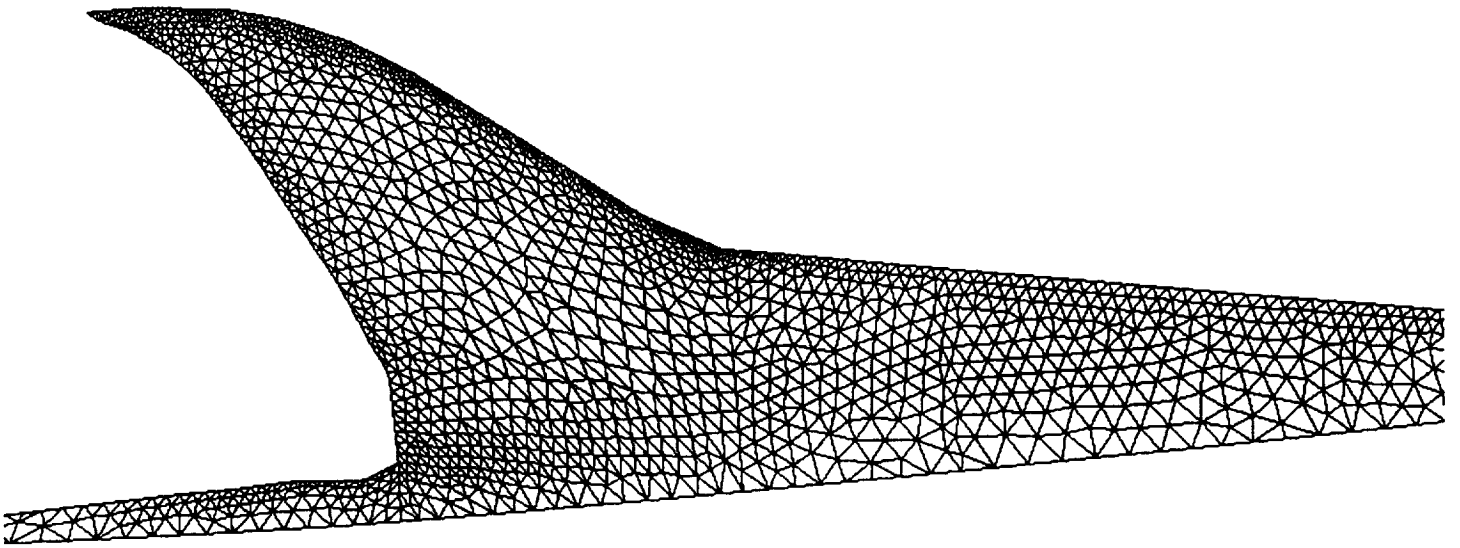
FIGURE 3. Portion of grid on upper surface of HSCT.
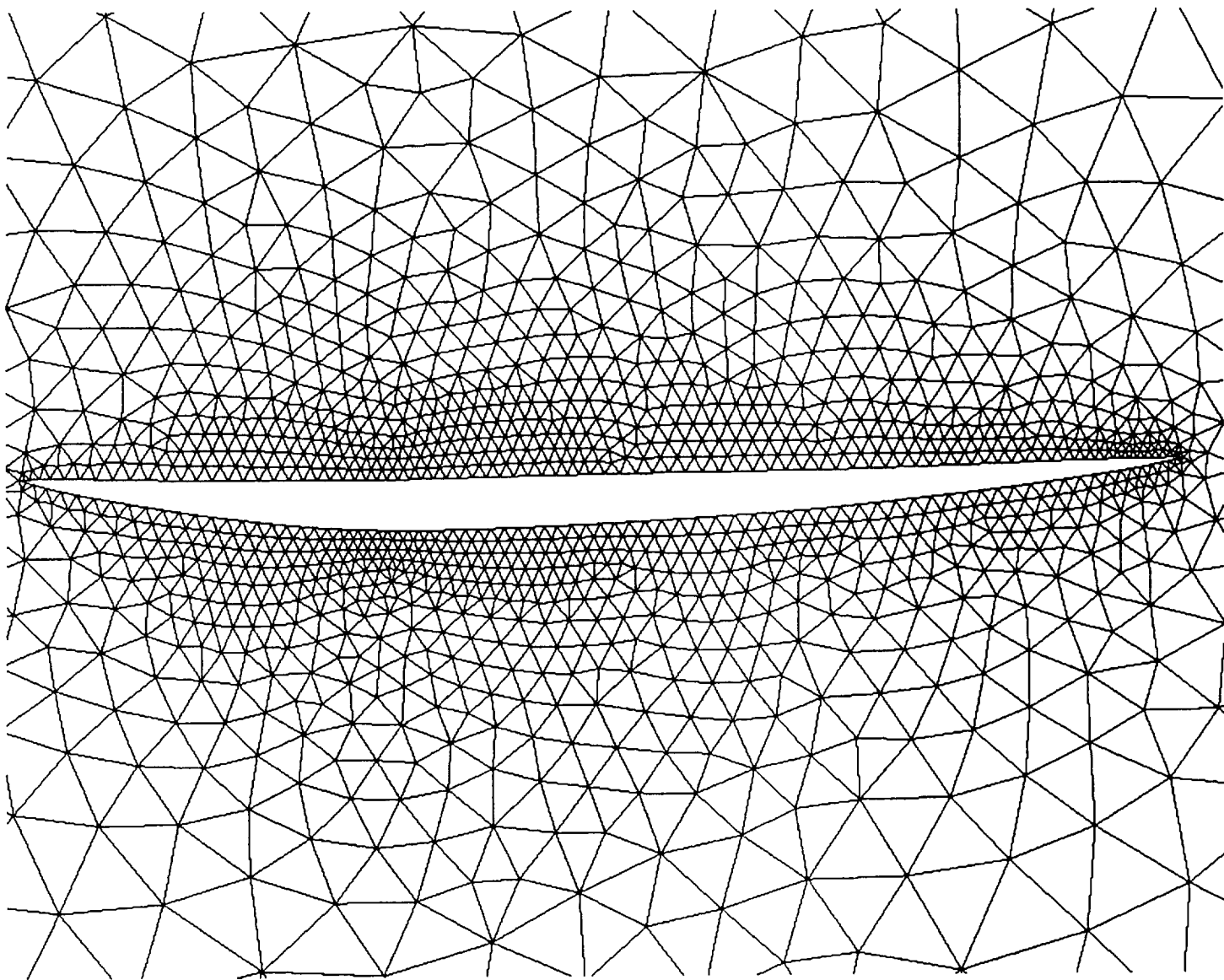
FIGURE 4. Portion of grid on lower surface of HSCT.

FIGURE 5. Portion of grid in plane of symmetry for HSCT.

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1993	Contractor Report

4. TITLE AND SUBTITLE
VGRIDSG - An Unstructured Surface Grid Generation Program

5. FUNDING NUMBERS
C NAS1-19038
WU 505-90-53-02

6. AUTHOR(S)
Michael J. Bockelie

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Sciences Corporation
3217 N. Armistead Ave.
Hampton, VA 23666-1379

8. PERFORMING ORGANIZATION REPORT NUMBER
TAO 60291

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-191426

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Samuel A. McPherson ACD-CMB

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
This report contains an overview of the VGRIDSG unstructured surface grid generation program. The VGRIDSG program was created from the VGRID3D unstructured grid generation program developed by Vigyan, Inc.. The purpose of this report is to document the changes from the original VGRID3D program and to describe the capabilities of the new program.

14. SUBJECT TERMS
Surface Grid Generation; Unstructured Grids; Parametric Space Representation

15. NUMBER OF PAGES
14

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	