

S3D - AN INTERACTIVE SURFACE GRID GENERATION TOOL*

Raymond Ching-Chung Luh
MCAT Institute
Moffett Field, CA

Lawrence E. Pierce
Sterling Software
Moffett Field, CA

David Yip
NASA Ames Research Center
Moffett Field, CA

SUMMARY

This paper describes S3D, an interactive software tool for surface grid generation. S3D provides the means with which a geometry definition based either on a discretized curve set or a rectangular point set can be quickly processed towards the generation of a surface grid for CFD applications. This is made possible as a result of implementing commonly encountered surface gridding tasks in an environment with a highly efficient and user-friendly graphical interface. Some of the more advanced features of S3D include surface-surface intersections, optimized surface domain decomposition and recombination, and automated propagation of edge distributions to surrounding grids.

INTRODUCTION

Advances in computer technology and Computational Fluid Dynamics (CFD) in the recent years have made possible the computation of flow fields around realistic three-dimensional geometries. One problem with realistic geometries is that they tend to be complex as well. No longer are these geometries defined by simple, analytic forms as was usually the rule in the past. As a result, the task of generating a computational grid, which is the first step in any CFD analysis, has become so labor-intensive and time-consuming that it can no longer be taken for granted. The question is not whether a grid can be generated but rather how long the process would require.

A computational grid is the collection of a large number of points about or within an object. Of particular importance are those points which lie directly on the surface of the object to help define its shape. Surface grid generation, a prerequisite to volume grid generation, is seen to have a dominant effect on the quality of the volume grid and to be extremely time-consuming. S3D is a new and evolving surface grid generation tool based on the work described in Refs. 1 and 2 that promises to significantly reduce the turn-around time.

* Dr. Luh's participation in this project was under cooperative agreement to MCAT Institute, NCC2-513, and Mr. Pierce's participation under contract to Sterling Software, NAS2-13210.

The program is implemented in the dynamic color-graphics environment of a workstation to take advantage of the latest in interactive technology. Much attention has been given to the issues of ease of use and user-friendliness in the design of the graphical interface for S3D. Considering the highly visual nature of the surface grid generation process, every effort has been put into facilitating the use of the mouse instead of the keyboard for maximum efficiency in interactivity. Thus, the user is able to perform nearly all of the tasks, including object manipulation, by moving the mouse around and clicking on the appropriate buttons. Windows for custom panels are opened and closed automatically as the need for them comes and goes so that the screen always displays just the right amount of information for the task at hand. A typical S3D layout is shown below.

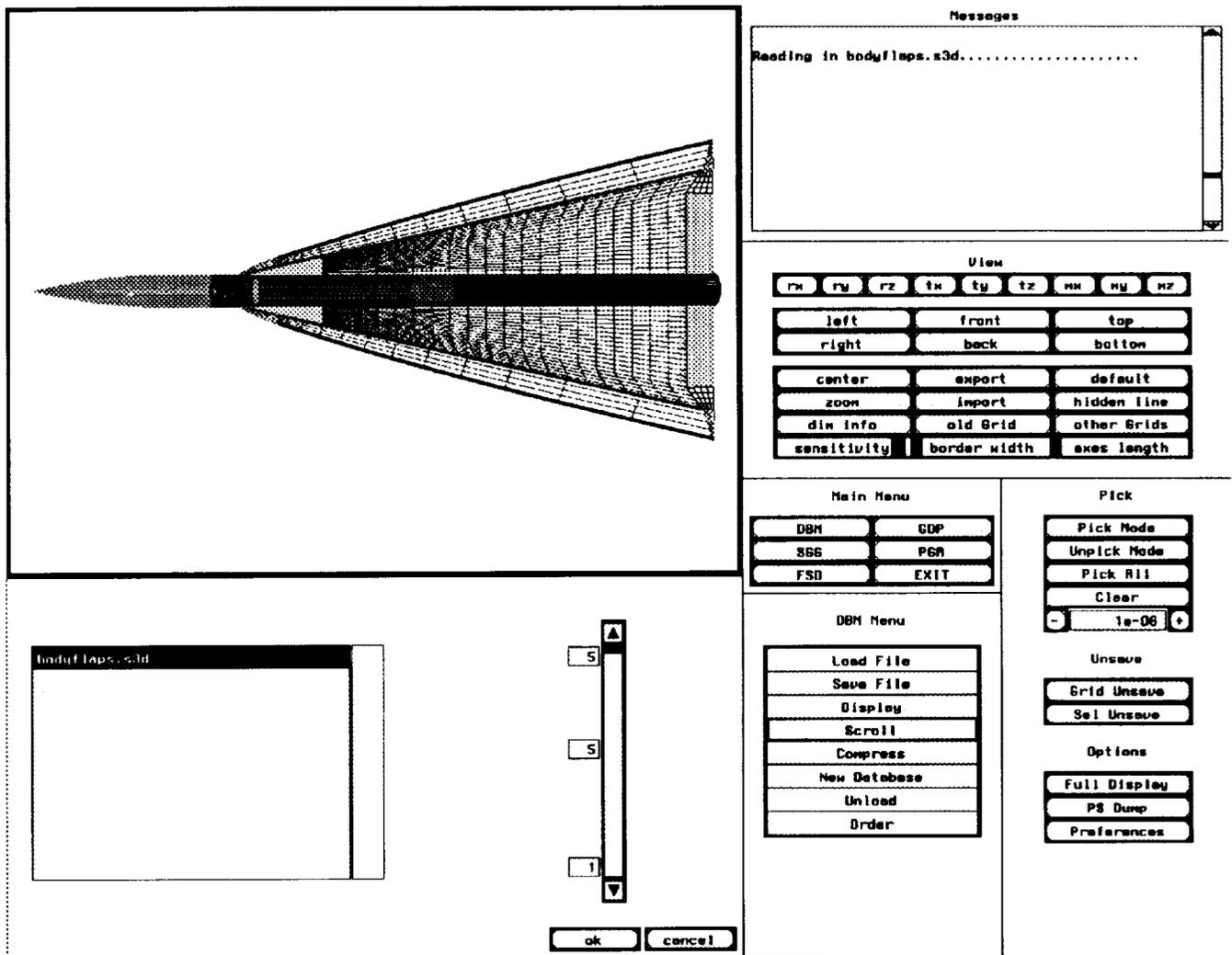


Figure 1. Typical screen layout of S3D

There are five major elements to S3D - Data Base Manager (DBM), Patch Grid Assembler (PGA), Geometry Data Processor (GDP), Surface Grid Generator (SGG), and Free-form Surface Designer (FSD). DBM controls the functions that allow the user to manipulate the S3D database, such as loading and saving

files, choosing which loaded files to use, and keeping track of the history of the database development. Functions in GDP permit processing of geometries made up of either a discretized curve set or a rectangular point set. A surface definition in the form of a reference grid can be obtained. In the process, data points which highlight geometrical discontinuities and high-curvature regions can be frozen to preserve the prominent features they represent. SGG functions work on a reference grid, or a rectangular point set. This is where surface grids can be customized. Work is under way to enhance this module so that geometry data from CAD systems can be directly input and analyzed without any approximation. PGA is a utility module that serves a variety of needs during the course of a surface grid generation operation. Deficiencies in the original geometry, if present, can be corrected or eliminated. Decomposition and recomposition of surface patches are made simple. The FSD module, which is still in the conceptual stage, will contain functions for designing surfaces which are necessary in the construction of faces, other than those given, for volume grid generation. Details regarding some of the more important functions in these modules are described below. A complete description of all existing functions can be found in Ref. 3.

DATA BASE MANAGER (DBM)

It is empirically evident that ease of input and output operations, and also the ability to retain and manage the database in hierarchical as well as cross-relational senses are vital to a patch-oriented system like S3D. DBM is the module in S3D that handles these functions in a menu-driven environment.

The input and output operations presently support three basic file types. Each file type can be manifested as a formatted (ASCII) file, an unformatted file, or a binary (C-binary) file. The three basic file types are the multi-grid PLOT3D⁴ format, a section data format, and a S3D historical format. The PLOT3D format, which has grown in popularity as a means for transferring data between various codes in the CFD processes, allows reading and writing of (x,y,z) data for one or more rectangular patches. This format is best suited for an existing surface definition or grid.

The sectional format is meant for surface data that is represented by a set of randomly distributed sectional points. At present, this format also serves as a buffer between the surface definition that might be stored in a CAD system and one that would be constructed using GDP. A plan to incorporate some of the more popular CAD standard formats such as IGES is being considered for future implementation.

The S3D historical format allows the surface data to be input or output with a large quantity of other information that is associated either with the geometry or with the hierarchy and the relationship of one patch to another. This format would be the best choice during the surface grid generation process because it allows a maximum level of information to flow in and out of S3D.

The broad historical information that is part of the S3D internal data structure permits DBM to feature a wide range of data management functions. It allows the user to make some data sets, which are typically collections of patches from a configuration, active (i.e. displayed) and make other data sets inactive (in memory, but not currently displayed). In addition, the user can select those data-objects such as patches, sections, and points on which to apply some function of S3D (e.g. redistributing patches, inserting points, etc.).

DBM also allows the user to manipulate the data in a historical fashion. A one-step "undo" feature makes it possible for the user to undo the previous operation. The user can also do a one step undo on any patch (even one that was not involved in the previous operation) or collection of patches at any time. The

user can also recall patches from any historical level, e.g. ten steps back, or construct new data sets using patches from various historical levels.

PATCH GRID ASSEMBLER (PGA)

This module is necessitated by the variety of cutting and pasting needs that arise as a result of the grid generation requirement or the way in which a geometry is given. The graphical interactive approach of PGA provides an easy and efficient way to accomplish this simple but bothersome task. The patch boundaries function is one of the most important tools in this module. It conveniently simplifies decomposition and recomposition of surface patches.

The add and change point functions allow the user to type in data points that are known to be missing or to replace bad existing ones. The delete points function allows the user to remove spurious or bad data points. There are also functions for manipulating how the geometry is stored with respect to the I and J axes. The user can swap the axes as well as reverse their direction.

The collapse edge feature enables the user to set all the points along an edge to the same coordinates. One use for this would be to correct small inconsistencies in the geometry being processed. For example, the coordinates of the points making up a collapsed point may be different by a small amount. This function would allow all the points to be set to the same coordinates.

Finally, a mirror function allows the user to create a new patch that is simply a reflection of the existing geometry. The reflection is made along either the x, y or z axis. The entire geometry can be reflected or just a portion of it. This may be useful if, for example, only half of a symmetric geometry was defined originally. The other half can be easily generated.

GEOMETRY DATA PROCESSOR (GDP)

From practical experience, the geometry data one obtains for a configuration may often be at a primitive level, i.e. with only a preliminary surface definition. The fuselage in such a geometry definition, for instance, may be given in cross sections with each section represented by a different number of points. There may be geometrical discontinuities or body-embedded singularity points which would require special attention. Usually, the work required to process such data outside of a surface grid generation package is partly to blame for the long delay in getting the surface grid.

GDP works like a section data editor allowing primitive point data to be processed into a surface definition based on a rectangular grid. Along the way, it permits domain decomposition, associated with the composite block-structured approach necessary for complex geometry, to be carried out on the surface. The key function under this menu is the redistribution of sections. The starting point for this operation is a set of cross sections, each with an arbitrary number of points distributed randomly. Each curve data is fitted with piecewise parametric cubic polynomials. This involves the determination of derivatives at each data point which is most critical in ensuring the accuracy of interpolation. A method adapted from Akima's scheme⁵ has shown to be highly accurate because of its second-order nature and independence of interval widths in the data. It should be pointed out that one may opt to redistribute points in the uni-variate sense of GDP even if the geometry data in question is rectangular. Uni-variate redistribution is preferred when

one is satisfied with the point distribution in one direction and wishes to preserve the positioning and shapes of those curves.

Not unexpectedly, geometry defined by cross sections could be missing data points that may be critical in outlining prominent geometric features in the opposing direction, i.e. in the direction of the cross sections. These are usually missing because they are not needed for adequately defining the cross-sectional curves. For such deficiencies, the global/local insert functions under GDP become rather convenient in allowing point insertions through interpolated means. The Break-line feature of GDP is another function that is essential in preserving prominent geometric features in the cross-sectional curves. With this function, one can essentially freeze any point(s) on the curves to keep them invariant under redistribution. Figure 2 below illustrates the results of carrying out the uni-variate redistribution on all curves to obtain the same number of points while preserving sharp corners and clustering around high-curvature regions.

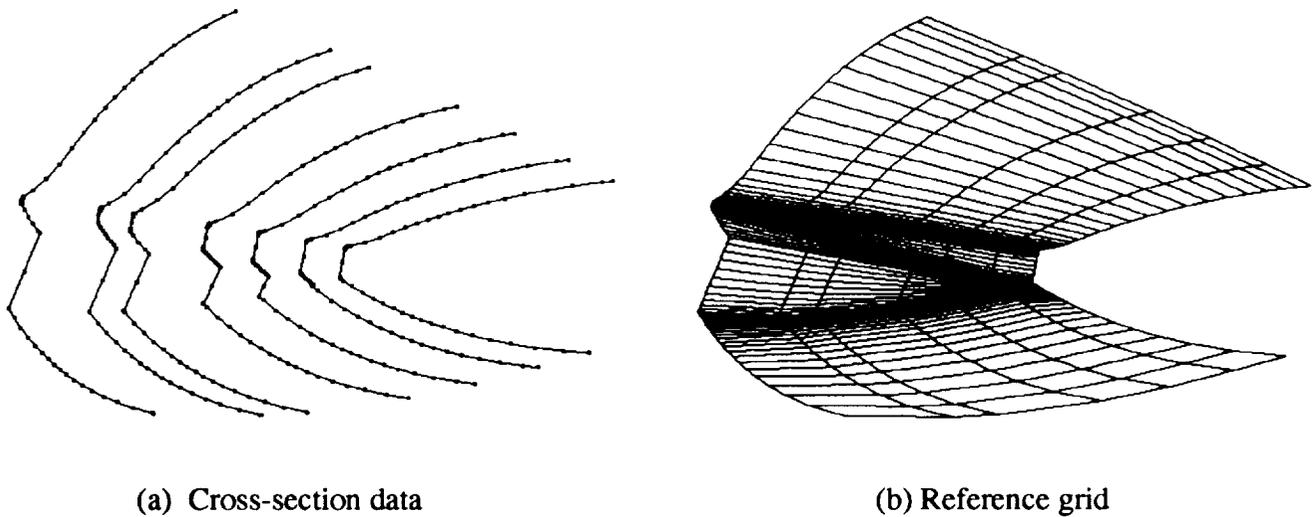


Figure 2. Cross sections with different numbers of points are first turned into a reference grid by applying GDP's uni-variate redistribution function

SURFACE GRID GENERATOR (SGG)

The output of GDP is a set of surface patches which clearly defines the surface geometry and is ready for surface grid generation. As redistribution in GDP is confined to the individual sectional curves, the bi-directional redistribution of these patches becomes the central task in SGG. One can think of GDP as making the transition from one form of surface definition (cross-sectional data) to another (reference grid). As explained above, one can redistribute a rectangular grid in the uni-variate sense using the functionality provided in GDP or, as is more often the case, in the bi-variate sense of SGG described below.

The first step in the redistribute function of SGG is to fit the grid points with piecewise bi-cubic patches based on a Pseudo Normalized Arc-Length Parameter Space (PNALPS) parametrization, which maps the tri-variate surface to a bi-variate unit square. Again, the critical derivative information is calculated at each

grid point using a method adapted from Akima's scheme.⁶ The desired point distribution in terms of PNALPS parameters is then set up interactively by the use of hyperbolic-function-based, two-sided stretching functions developed by Vinokur.⁷ Finally, the coordinate values corresponding to the new PNALPS are obtained by interpolating on the bi-cubic patches. Detailed descriptions are found in Ref. 1.

One highly significant feature of S3D is its ability to set up patch neighborhood connectivity information automatically. It cannot recognize staggered neighboring patches but it can contend with the multiple neighbor scenario if there is point continuity among all the patches at their corners. The user is able to pick one or more (or even all) of the patches for redistribution as long as they are valid neighbors in the sense of being topologically rectangular. Users never need be concerned with the connectivity or orientation of the patches. The result of such a redistribution is one single patch. Another equally powerful feature of S3D, made possible by the automatic patch connectivity capability, is the ability to export point distributions to neighboring patches at the touch of a button. This permits any local modification on the surface grid to be immediately reflected on the surrounding regions without additional effort.

Another frequently encountered problem tackled in SGG is the surface-surface intersection. Given two patches which intersect each other, each patch alternates between being a target and a source. An intersection curve is calculated for each source patch from the points where its grid lines intersect the target surface. The patches can then be redistributed further for point and slope continuities at the intersection or patch boundaries.

One frequently encountered task in surface grid generation is the changing of grid density. This seemingly simple task can nevertheless be annoyingly time-consuming, especially when the geometry in question contains many features of physical significance. SGG is perfectly suited for such a task because of the ease with which existing grids can be redistributed. Figure 3(a) shows the space shuttle forebody represented by a single grid of dimension 66 by 77. In very little time, a new grid of dimension 48 by 60, which is a reduction of nearly 50% from the original, is obtained as shown in Figure 3(b). The less dense grid is generated such that grid point concentration near and around all major feature lines in the shuttle geometry is main-

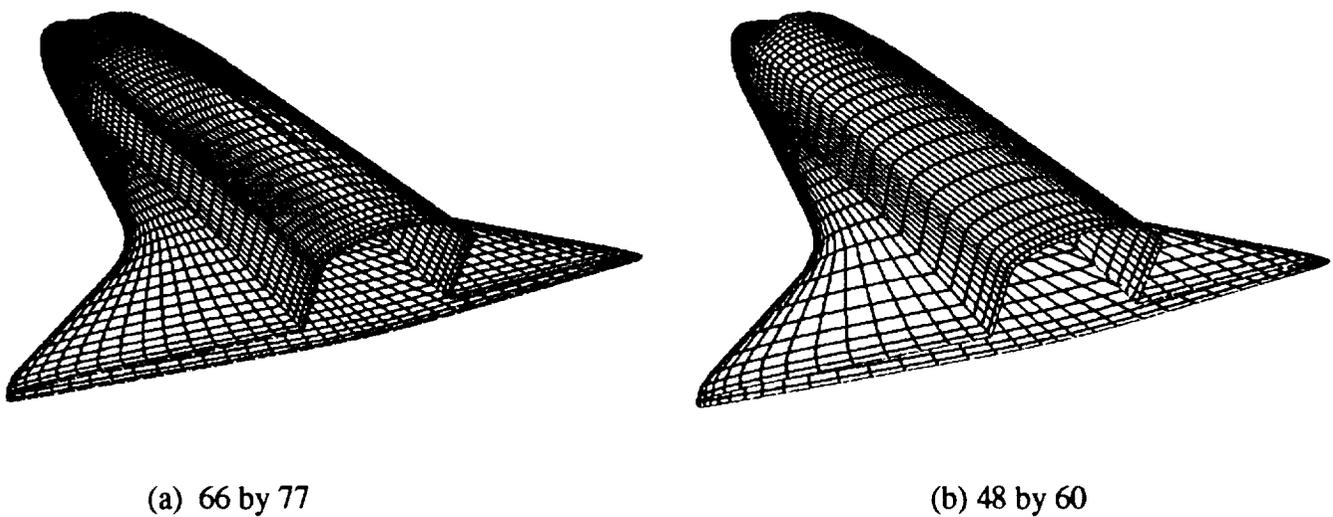


Figure 3. The grid in (b) is obtained by reducing the density and redistributing the grid in (a)

tained. This is made possible by first decomposing the original single grid into multiple grids or patches, then redistributing each patch and finally recomposing the multiple grids back into one piece.

The similarity in grid point concentration for the two grids above can be appreciated by looking at the PNALPS before and after the redistribution as shown in Figure 4(a) and 4(b), respectively. The blunt nose is on the left edge of the parameter space and the coarser grid is designed with more grid point concentration locally. The streak in the middle running down to the right side is where the wing leading edge is. Note that the coarser grid does not have the spacing discontinuities which are quite visible in the original.

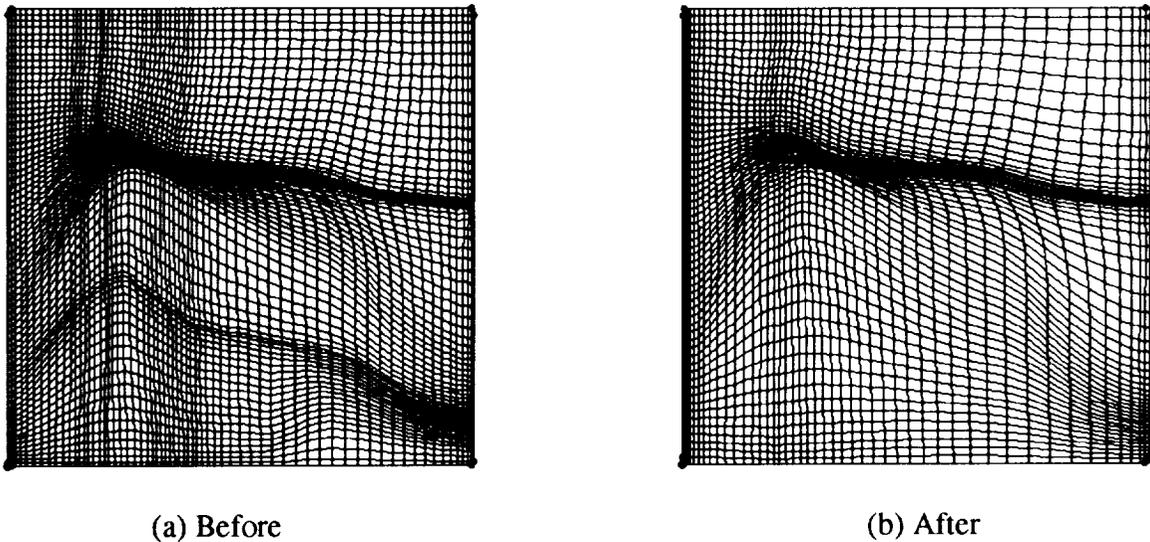


Figure 4 Comparison of PNALPS before and after grid density reduction on the shuttle body (fig. 3)

CURRENT AND FUTURE EFFORT

At present, S3D can start with a geometry definition based either on a discretized curve set or a rectangular point set as described above. There is however an increasing awareness in the CFD community of the need to preserve and utilize Computer-Aided-Design (CAD) representations throughout the geometry modeling and surface gridding process, so that the resulting grid points stay true to the exact definition. In order for a surface modeling tool such as S3D to fulfill this requirement, it must be modified to accept geometry data directly from the CAD systems, and to analyze such data without any approximation. As a first step, work is under way to add CAD data input to the existing I/O capabilities of S3D. Most existing CAD systems output data using Initial Geometry Exchange Standards (IGES), and hence the proposed input function would be one that reads CAD data in IGES format. In particular, the CAD input function will adhere to NASA-IGES⁸ which is an emerging geometry data exchange standard based on a subset of IGES specifically designed for CFD applications. NASA-IGES includes formats for representation of curves and surfaces by Non-Uniform Rational B-Splines (NURBS) which are fast becoming the industry standard for surface designs and analyses.

The implementation of the proposed CAD input function would enable direct down-loading of surface geometries designed by CAD systems into S3D where the data would be stored in its exact form. One approach being considered for generating the surface grid for this high-order geometry definition is as follows. It is proposed that a reference grid that would lie exactly on the true surface be extracted from each CAD surface. Slope and curvature information of the CAD surfaces would be used to determine the distribution of points on the reference grids. At this point, the present capability of S3D in bi-variate redistribution of grid points would be applicable. The resulting grids would then be projected onto the CAD surfaces to ensure accuracy.

CONCLUDING REMARKS

An overview of S3D, a new and evolving surface grid generation tool, has been presented. Some of the more useful features and capabilities of the program have been described. When completed, S3D will be able to process geometry definitions based on discretized curve sets, rectangular point sets, or CAD surface representations. Creative use of state-of-the-art interface technology together with innovative algorithms designed to automate repetitious and labor-intensive tasks makes S3D an indispensable as well as a convenient tool. The efficiency and ease with which S3D can be used to handle commonly encountered tasks in surface grid generation are already helping CFD analysts to drastically reduce the time required in gridding complex surface geometries.

REFERENCES

1. Luh, Raymond C.-C.: Surface Grid Generation for Complex Three-Dimensional Geometries. Appears in Sengupta, S.; Häuser, J.; Eiseman, P. R.; and Taylor, C., eds.: *Numerical Grid Generation in Computational Fluid Mechanics*. Pineridge Press Ltd., 1988.
2. Luh, Raymond C.-C.; Yip, David; and Pierce, Lawrence E.: Interactive Surface Grid Generation. Appears in Arcilla, A.S.; Häuser, J.; Eiseman, P.R.; and Thompson, J.F., eds.: *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*. North-Holland, 1991.
3. Luh, Raymond; Pierce, Larry; and Yip, David: *S3D User Manual*, Version 1.0, 1991.
4. Walatka, P. P.; Buning, P. G.; Pierce, L.; and Elson, P.A.: *PLOT3D User's Manual*, Version 3.6. NASA TM-101067, 1990.
5. Akima, H.: A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *J. of the ACM*, Vol. 17, No.4, 1970, pp. 589-602.
6. Akima, H.: A Method of Bivariate Interpolation and Smooth Surface Fitting Based on Local Procedures. *Comm. of the ACM*, Vol. 17, No. 1, 1974, pp. 18-20.
7. Vinokur, M.: On One-Dimensional Stretching Functions for Finite Difference Calculations. *JCP*, Vol. 50, No. 2, 1983, pp. 215-234.

8. Blake, Matthew: The NASA-IGES Geometry Data Exchange Standard. *NASA Workshop on Software Systems for Surface Modeling and Grid Generation*, NASA CP-3143, April 1992.

