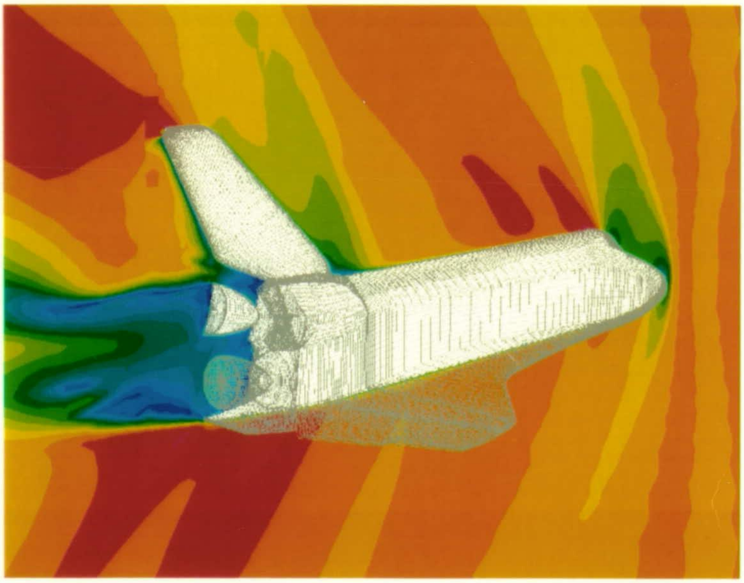
MCAT Institute
Progress Report
93-05

Numerical Computation of Complex Multi-body Compressible Navier-Stokes Flows with Applications for the Integrated Space Shuttle Vehicle

William M. Chan



February 1993

NCC2-654

MCAT Institute
3933 Blue Gum Drive
San Jose, CA 95127

CASI

Numerical Computation of Complex Multi-body Compressible Navier-Stokes Flows with Applications for the Integrated Space Shuttle Vehicle

William M. Chan

Abstract

An enhanced grid system for the Orbiter was built by integrating CAD definitions from several sources and then generating the surface and volume grids. The new grid system contains geometric components not modelled previously plus significant enhancements on geometry that has been modelled in the old grid system. The new Orbiter grids were then integrated with new grids for the rest of the launch vehicle.

Enhancements were made to the hyperbolic grid generator HYPGEN and new tools for grid projection, manipulation and modification, Cartesian box grid and far field grid generation and post-processing of flow solver data were developed.

Introduction

This project forms part of the long term computational effort to simulate the time dependent flow over the integrated Space Shuttle vehicle during its ascent mode for various nominal and abort flight conditions. One of the important parts of this project is to obtain better comparisons between numerical simulations and flight data. In the simulations performed so far, the geometry has been simplified in various ways to reduce the complexity so that useful results can be obtained in a reasonable time frame due to computer resources limitations. With the ICEM CAD/surface grid generation software, the surface definitions of different components of the Space Shuttle are now more accurately represented. In this project, surface and volume grids with more refined details are generated for the Orbiter. The complex process of inter-grid connectivity is then performed using the PEGASUS code. The robustness of the new grids in the Orbiter base region is tested by first combining with the old Orbiter fuselage and wing grid and performing a flow simulation for an Orbiter alone case. The enhanced new Orbiter grid system is then integrated with the enhanced External Tank, Solid Rocket Boosters and attach hardware grid systems developed at NASA-Johnson to form the enhanced Space Shuttle Launch Vehicle (SSLV) grid system.

In the past year, the hyperbolic grid generation code HYPGEN has been modified significantly to achieve a more modular structure. The algorithm has also been improved to provide robustness for a wide range of different geometries [1]. This project includes further research in developing even more robust algorithms. The ultimate goal is for the code to be able to intelligently select as many input parameters related to grid quality as possible. Thus, the user input will be reduced to a minimum.

Various useful tools are developed to improve and speed up the process of grid generation and manipulation and flow solution data post-processing. These include improvements made to the grid projection tool PROGRD used to project subsets of an active grid on to subsets of reference grids. This is an essential step in the Chimera overset-grid approach for inter-grid connectivity at the body surface. A grid editor GRIDED is developed that includes many common operations performed on a single block structured grid into a unified program. Another program called BOXGR is developed to generate the Cartesian box grids and far field grid used in the Chimera overset grid approach (see section on BOXGR below). A post-processing tool OVERPOST is developed to convert the output data files generated by the flow solver OVERFLOW [2] into a format suitable for plotting using XYPLOT.

Algorithm and Code Development

Hyperbolic Grid Generator - HYPGEN

Further optimization was performed on the code including the development and implementation of more efficient block tridiagonal solvers resulting in a 10% speed up of the code. Slight algorithm improvements were made for the sharp convex corner logic and floating boundary conditions at singular axis points. Version 1.2 of the code was released in March, 1992 to users in the RF Division at Ames and to the Shuttle group at NASA Johnson. In 1992, the code was requested by and distributed to the following groups: Rockwell International, Nielson Engineering, Computational Fluid Dynamics Applications, Penn State University, The Boeing Company, Wright Lab., Boeing Defense and Space Group, Office of Aircraft Compatibility (Dept. AF), Douglas Aircraft, NASA Langley, Bell Helicopter, B-2 Division (Northrop Corp.). A detailed description of the algorithm for HYPGEN is given Appendix A.

Grid Projection Tool - PROGRD

This program reads in an active grid and performs projections of subsets on the body surface (L=1 shell) of the active grid on to subsets on L=1 of one or more reference grids. Projections can be performed along surface normals or along any of x, y or z directions.

PROGRD is required where solid surfaces from two grids overlap. This occurs not only in the overlapped regions between a collar grid and its parent grids but also in complex overlapped grid regions such as the backend of the orbiter. There are two main areas where PROGRD is required:

(1) The outer boundary of one grid has to lie on the bilinear surface of the neighboring grids for PEGASUS to find the right interpolations, e.g., the outer boundary of the collar grid has to be projected on to the L=1 surfaces of the parent grids.

(2) Hole boundary points of one grid has to lie on the bilinear surface of a neighboring grid for PEGASUS to find the right interpolations, e.g., the hole boundary points of a parent grid have to lie on the L=1 surface of the collar grid. Version 3.2 or later of

PROGRD has the added ability to look for fringe orphan points with an iblank value of 101 (compatible with latest PEGASUS 4.1 +).

The later versions of PROGRD (3.0 or later) allows the recording all projections that have to be performed for the active grid under a single input file. Not all subsets recorded in the input file are projected. Projections are performed only on the subsets whose reference grid file can be found under the specified directory. This allows continuous modifications of the active grid without keeping the old reference grids around. Version 3.1 or later also accepts a different mode and projection direction for each subset.

Grid Editing Tool - GRIDED

This is an effort to include many common operations that are performed on a grid into a single code. Separate programs are in existence to perform each of these tasks but it is cumbersome to have to carry all the different programs around.

The GRIDED program can be used to perform the following common operations on a 2D/3D single grid file in PLOT3D format. The operations can be performed in sequence one after another. After each operation, a warning message is printed if the original grid has changed from a right-handed to a left-handed system or vice versa. The grid indices are labelled as J, K and L.

Operations that do not change the grid dimensions include

(1) Convert to/from formatted/unformatted, planes/whole, 2D/3D data types
(2) Interchange J and K grid families
(3) Interchange K and L grid families
(4) Interchange J and L grid families
(5) Reverse index direction in J and/or K and/or L
(6) Scale and translate
(7) Mirror about x=0, y=0 or z=0 plane

Operations that change the grid dimensions include

(1) Extract a subset
(2) Add any number of extra planes in J, K or L by extrapolation
(3) Add extra planes in J or K by reflected symmetry
(4) Add extra planes to 2D grid to form 3D grid for '2D' or 'axisymmetric' option in OVERFLOW
(5) Duplicate and mirror reflected symmetry grid to form periodic grid
(6) Read in a new grid and concatenate to current grid

Box and Far Field Grid Generation Tool - BOXGR

In the Chimera overset grid approach for complex configurations, it is typical to grow the volume grids only a small distance out from each componenet and then embed the component volume grids in a Cartesian box grid. The Cartesian box grid typically consists of an uniformly spaced interior region which completely encloses the component volume grids, and outer layers where the grid spacing is stretched from the interior value to some desired value at the outer boundary. A tool called BOXGR was developed to generate box grids with arbitrary dimensions and grid spacings in the interior and outer regions. The same tool also has the capability to generate the inner surface of the far field grid where at least one point overlap is maintained with the outer boundary of the box grid. An ellipsoidal topology, built from smoothing the corners of the box grid, is used for the inner surface of the far field grid.

Post-processing Tool for OVERFLOW - OVERPOST

A post-processing tool named OVERPOST was developed for the Chimera overset-grid Navier-Stokes solver OVERFLOW. The function of OVERPOST is to extract appropriate data from the output files from OVERFLOW for plotting purposes using XYPLOT. Processed files are generated for histories of conservative variables residuals, forces and moments, minimums of pressure and density, turbulence model equations residuals and species residuals. A preference file can be initialized such that the user does not have to enter the same parameters each time OVERPOST is used.

Applications

The construction of the new enhanced Orbiter grid system is part of a team effort with NASA Johnson to enhance the grid system for the complete Space Shuttle Launch Vehicle. The first part of the work for this project is to generate surface grids for the Orbiter base region. Geometric components that were absent from the old grid system are now modelled. These include the body flap, the Space Shuttle Main Engines (SSMEs), the OMS extension and the RCS jet block. The surface geometry was obtained from Rockwell and ICEMCFD was used to manipulate the geometry and generate the surface grids. One region where geometry simplification was needed was in the RCS jet block where the CAD surface was modified to join smoothly with the OMS extension. Volume grids were generated using HYPGEN, and the inter-grid connectivity was performed using PEGASUS by combining the new base grids with the old Orbiter fuselage and wing grid resulting in an intermediate grid system consisting of 18 grids. The robustness of the new base grids was tested by running a flow solution using OVERFLOW with the intermediate 18-grid system (see Figure 1). Since the solution was fairly smooth in the base region, it was concluded that the topologies used for the new base grids were adequate to capture the flow in the region.

The next step was to generate enhanced grids for the Orbiter fuselage, wing and elevons. In the old grid system, the Orbiter wing and elevons were generated with some coarse approximations in places and no CAD definition was in existence at that time. Part of the effort in this project was spent on the construction of CAD surfaces for part of the wing from the Masters Dimensions Specifications [3]. During the course of the work, CAD surfaces for the wing tip, the elevons and parts of the wing were obtained from Kennedy Space Center and Rockwell. These latest CAD definitions were integrated together with the existing CAD model to form enhanced geometry definitions for the wing and elevons. A blocking strategy was developed for the model which replaced the single old Orbiter fuselage/wing grid with about 11 overlapped grids for the right half: Orbiter nose (ONOSE), Orbiter forward fuselage (ORB1), Orbiter back fuselage (ORB2), upper wing (WINGU), lower wing (WINGL), wing leading edge (WINGLE), elevons at 10/5 deflections (ELVS105), middle elevon gap (EGAPM105), in-board elevon gap (EGAPI10), Orbiter lower-middle fuselage (ORBLFM), Orbiter lower-back fuselage (ORBLFB) - see Figures 2 and 3. Introduction of the ONOSE grid removed the axis singularity at the nose. The WINGU and WINGU grids were designed to have grid lines alined with pressure taps where flight data is available for comparison with CFD results. The separate elevon and elevon gap grids allow different deflection angles to be modelled without changing the grids on rest of the Orbiter. A fine ORBLFM grid was used to enclose the complex grids for the ET/Orbiter attach hardware.

All volume grids were generated using HYPGEN except for the elevon gaps which were generated using ICEMCFD with a patched grid topology. The following box grids were generated using the BOXGR tool described above: a large box around the entire Orbiter with a 12 inch interior grid spacing and stretched to a 48 inch grid spacing at the outer boundaries, a small box around the Orbiter base grids with a uniform 5 inch grid spacing and a small box around the tip of the vertical tail. Inter-grid connectivity was performed using PEGASUS 4.1 for the right half of the Orbiter ($\approx$ 30 grids). The right half grids were then reflected to form left half grids. Some of the right and left grids were combined to form a single grid. Inter-grid connectivity was then performed for the complete Orbiter ($\approx$ 50 grids).

The final stage of the grid generation was to integrate the new enhanced Orbiter grid system with enhanced grids for the rest of the launch vehicle. Another box grid was generated to enclose the ET/SRB and the far field grid was also generated using BOXGR and HYPGEN. Slices of the SSLV volume grids, box grids and the far field grid on the symmetry plane are displayed in Figure 4. A 5-day meeting was held at Ames with the team from NASA-Johnson and Lockheed, Houston to integrate the enhanced SSLV grid system using PEGASUS 4.1. The final grid system contained 111 grids with 16 million points. More details of the development of this grid system are discussed in Appendix B.

Concluding Remarks

Useful tools were developed for grid projection (PROGRD), grid manipulation and modification (GRIDED), box and far field grid generation (BOXGR) and post-processing of OVERFLOW data files (OVERPOST). These tools have been distributed to users in the RF Division at Ames and to the Shuttle group at NASA-Johnson. Further optimization was performed on HYPGEN together with some slight algorithm improvements. HYPGEN has been distributed to many different groups in industry.

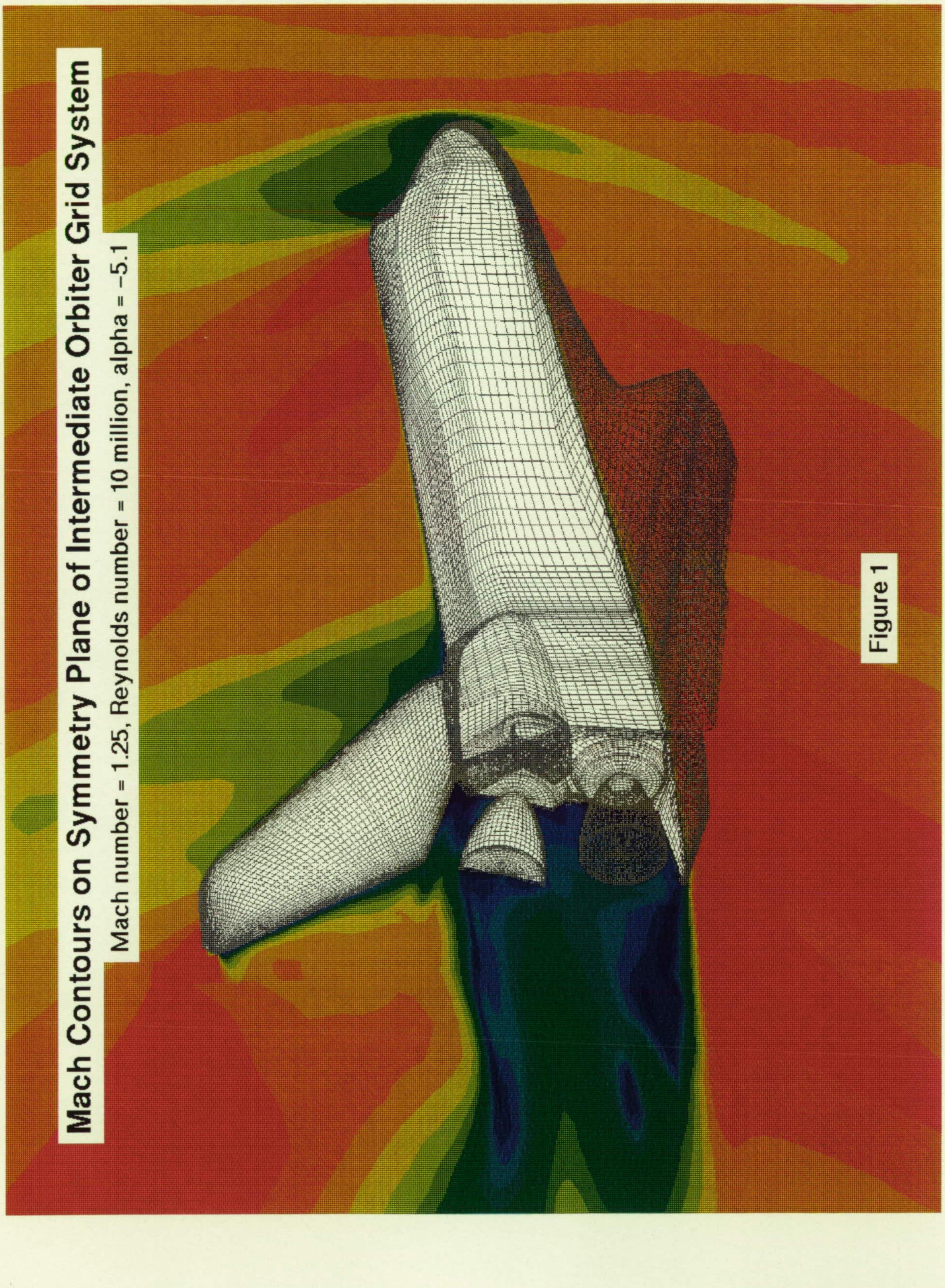
Enhanced CAD geometry definitions for the Orbiter obtained from several sources were combined into a single model. New grids were generated to model components absent from the previous grid system and to enhance resolution of existing components. The new Orbiter grids were integrated with new grids for the rest of the launch vehicle. The complete enhanced integrated SSLV grid system with 111 grids and 16 million grid points is one of the largest overset grid system built for complex aerodynamic configurations.

References

1. Chan, W.M. and Steger, J.L., "A Generalized Scheme for Three-Dimensional Hyperbolic Grid Generation", AIAA Paper 91-1588. Proceedings of the AIAA 10th CFD conference, Honolulu, Hawaii, 1991.

2. Buning, P.G., Chan, W.M., Renze, K.J., Sondak, D., Chiu, I.T. and Slotnick, J.P., *OVERFLOW User's Manual, Version 1.6*, NASA Ames Research Center, Moffett Field, CA, 1991.

3. "Space Shuttle Masters Dimension Specifications", Rockwell International MD-V70, Rockwell Space Systems Division, Downey, CA, Revised 1991.



Mach Contours on Symmetry Plane of Intermediate Orbiter Grid System

Mach number = 1.25, Reynolds number = 10 million, alpha = –5.1

Figure 1

ORIGINAL PAGE
COLOR PHOTOGRAPH

SURFACE GRIDS FOR NEW ORBITER GRID SYSTEM

FRONT VIEW

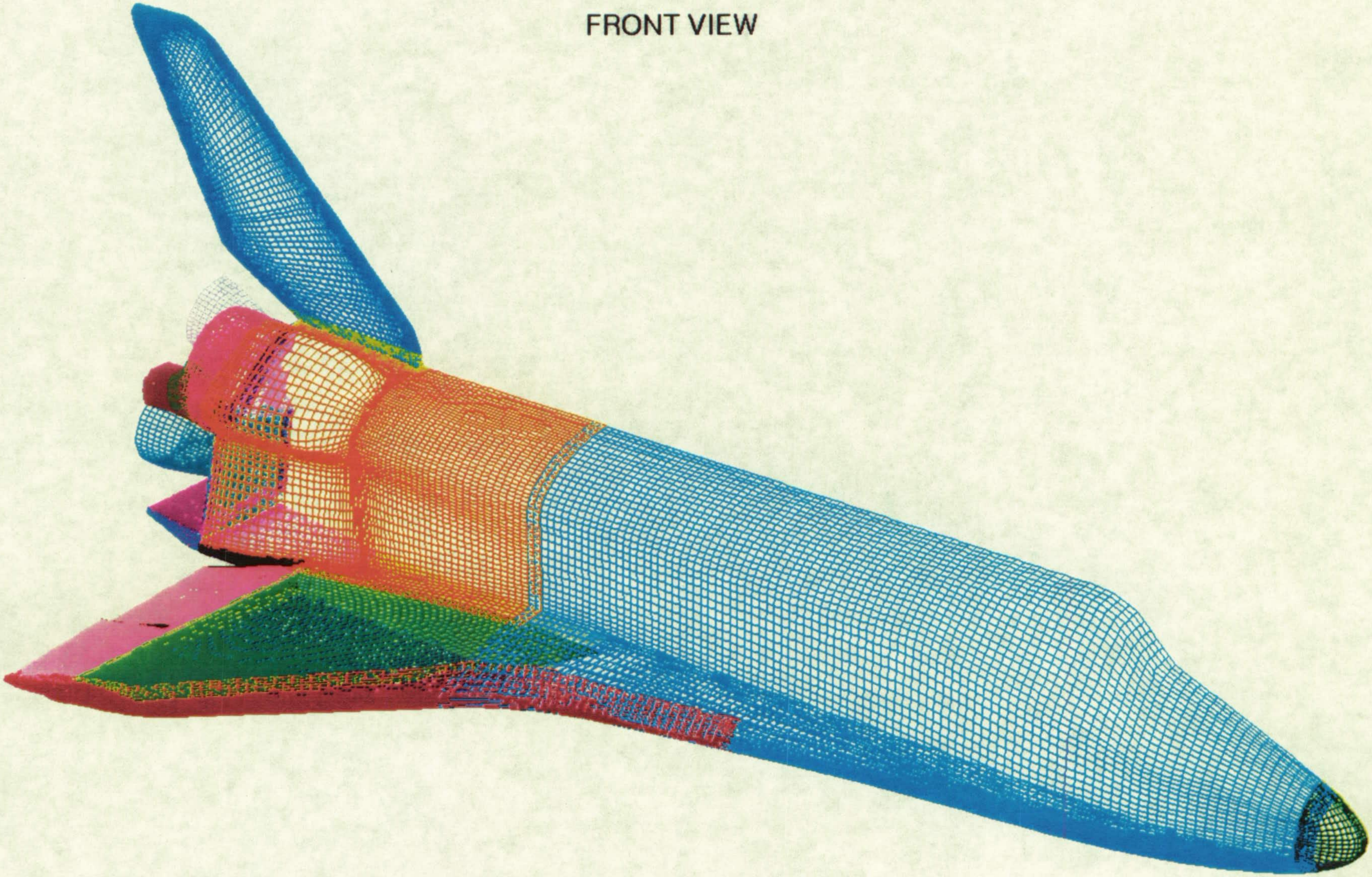


Figure 2

SURFACE GRIDS FOR NEW ORBITER GRID SYSTEM

Back View

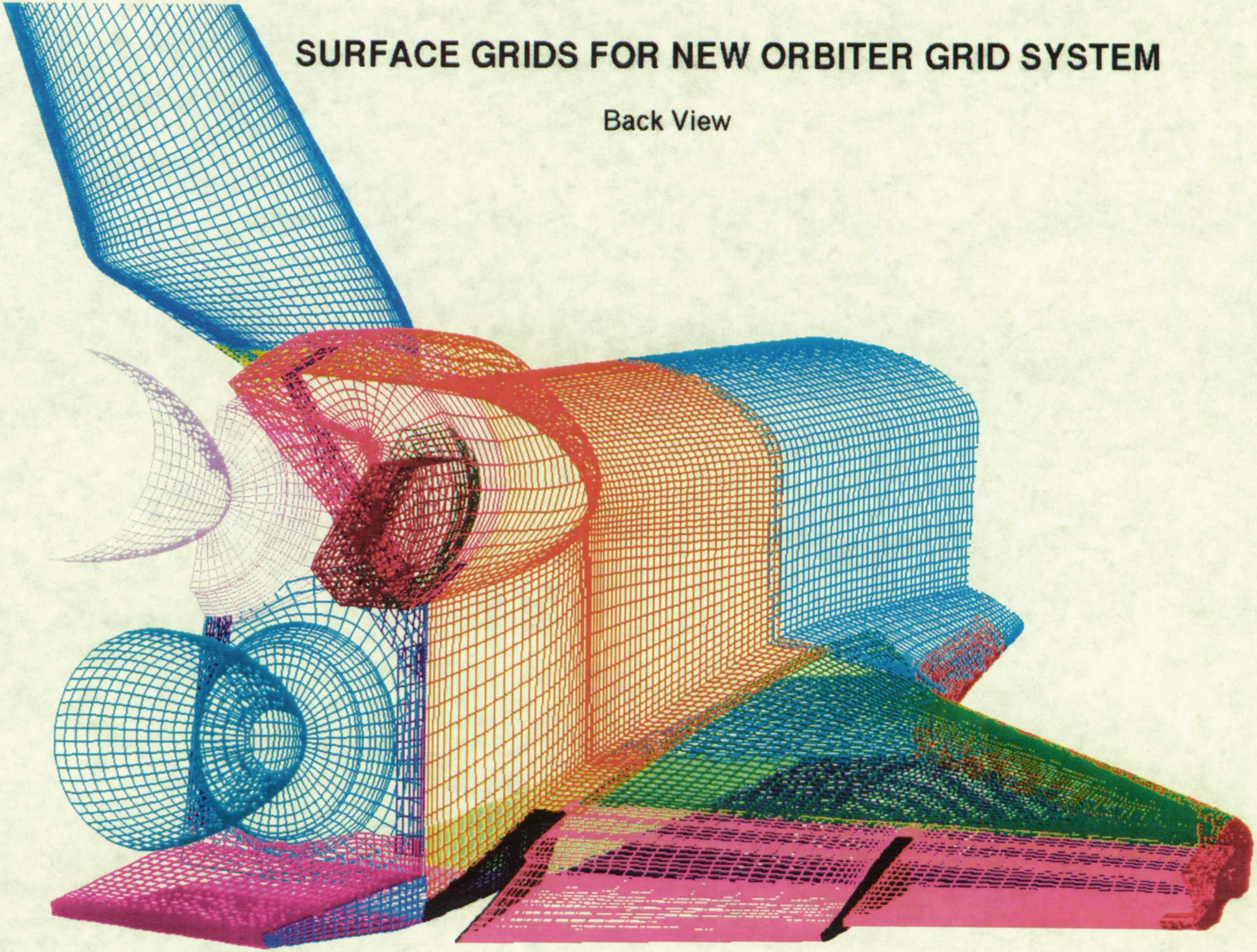


Figure 3

Symmetry Plane of Volume Grids for SSLV, Boxes and Far Field

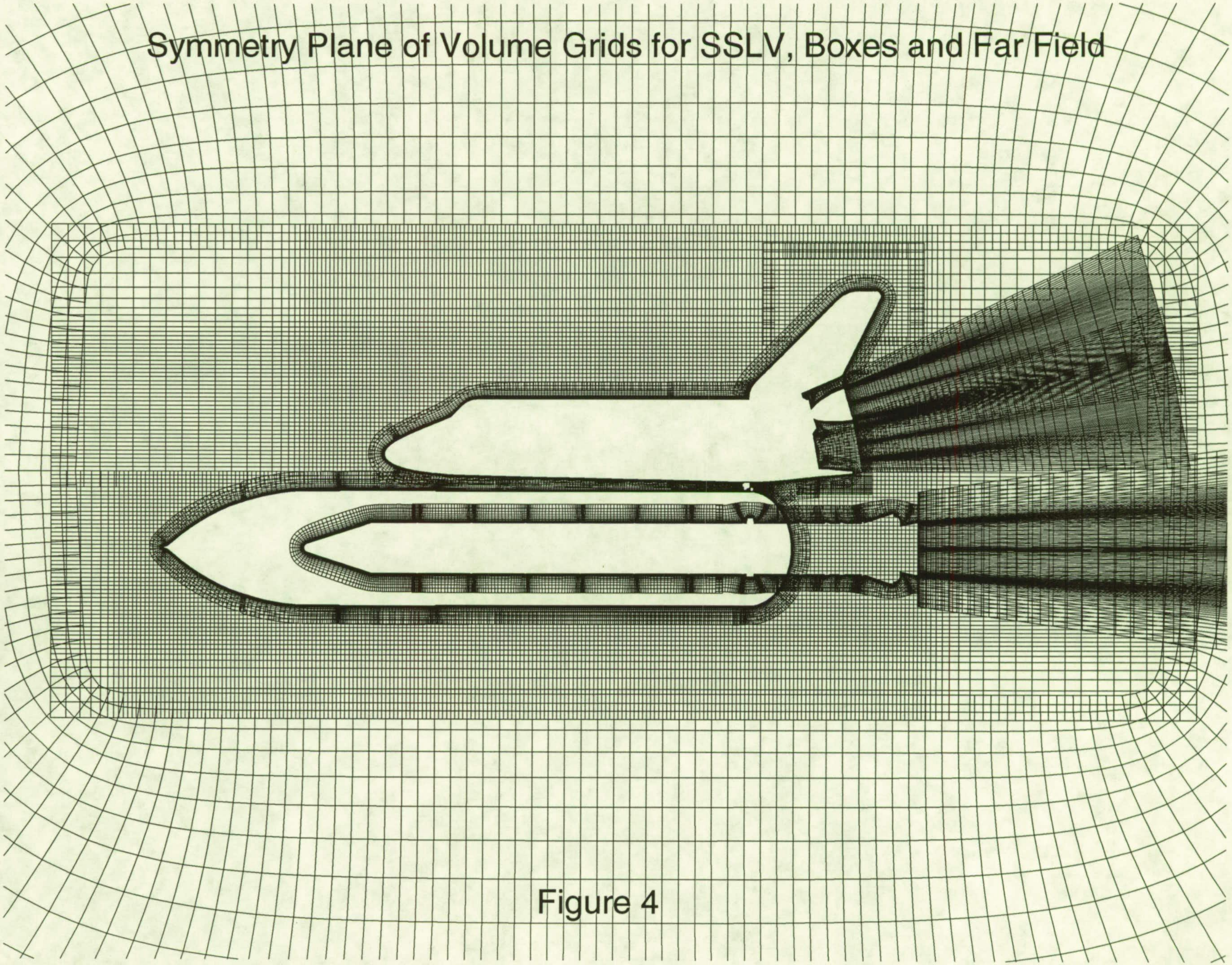


Figure 4

APPENDIX A

Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme*

William M. Chan

MCAT Institute
NASA-Ames Research Center
Mail Stop T045-2
Moffett Field, California 94035

and

Joseph L. Steger**

Department of Mechanical, Aeronautical and Materials Engineering
University of California at Davis
Davis, California 95616

Transmitted by Joe F. Thompson

ABSTRACT

A hyperbolic grid generation scheme formulated from grid orthogonality and cell volume specification has been significantly enhanced so that high quality three-dimensional grids can be obtained for a wide variety of geometries. While the speed of the scheme remains one to two orders of magnitude faster than typical elliptic grid generation methods, the robustness of the scheme has been greatly improved over previous applications of the three-dimensional hyperbolic grid generation procedure. Enhancements included the use of spatially-variable smoothing coefficient, metric correction procedures, local treatment of severe convex corners, and new extrapolation treatments of floating and axis boundaries. The versatility of the new hyperbolic grid generation scheme is demonstrated by three-dimensional grids generated for external components of the integrated Space Shuttle vehicle and the SOFIA telescope.

---

*The authors are grateful to Dr. Pieter Buning and Dr. Yehia Rizk for some very useful discussions. A special thanks is owed to Mr. Chris Atwood for providing the surface grid for the SOFIA telescope.

**Joseph Steger died on May 1, 1992; Computation Fluid Dynamics has lost a giant.

APPLIED MATHEMATICS AND COMPUTATION 51:181–205 (1992)

655 Avenue of the Americas, New York, NY 10010 0096-3003/92/$5.00

1. INTRODUCTION

One of the most popular approaches for generating structured grids is by the solution of a set of partial differential equations. The governing equations can be classified into three types: elliptic, parabolic and hyperbolic. The most widely used grid generation methods require the solution of a set of elliptic equations [4, 14–16]; however, parabolic and hyperbolic equations have also been successfully employed [3, 6, 8, 10, 12, 13] and are advantageous for certain applications.

Since the solution of the elliptic equations satisfies the maximum principle, the grids generated are typically smooth. Moreover, the formulation of the elliptic equations allows exact specifications of all boundary point locations. However, grid orthogonality cannot be maintained with conventional elliptic grid generation methods and boundary surface refinement can be difficult to impose. The user input required to set up boundary distributions can also be time consuming. Hyperbolic grid generation schemes produce nearly orthogonal grids, have excellent clustering control, and can be generated in one to two orders of magnitude less computer time than elliptic methods. However, hyperbolic grid generation methods are less robust, tend to propagate input discontinuities, and the outer boundary location cannot be precisely specified. Hence, they are usually restricted to the generation of grids for external flows or for chimera overset-grid schemes [2], where the exact location of the outer boundary is not constrained. This latter application has become more important, however, and this, coupled with the efficiency and often superior grid quality obtained with hyperbolic grid generation schemes, motivates investigation into improving their robustness.

In the last few years, the hyperbolic grid generation algorithm described in [13] has evolved to include some significant enhancements. A wider range of boundary conditions can now be treated and some feedback features have been added so that dissipative terms (which give the equations a somewhat parabolic nature and smoothness) are now automatically adjusted depending on grid evolution or character. The resulting hyperbolic grid generation scheme is significantly more robust, produces higher quality grids, and can treat a wider variety of topologies.

The governing equations for three-dimensional hyperbolic grid generation are presented in Section 2. The numerical marching scheme employed to solve these equations is described in Section 3. Three factors important in controlling grid quality, the boundary conditions, cell volume specification and added smoothing are discussed in Sections 4, 5 and 6, respectively. Discontinuities due to body shape or initial grid point distribution can present special problems to a hyperbolic solver that dissipation alone cannot satisfactorily cure. Corner points are especially difficult. A metric correction procedure which is essential in providing smooth grids at corners with uneven grid spacings is described in Section 7. Extra robustness at sharp convex corners can be achieved by switching from solving the hyperbolic equations to some other equations at the convex corner points. These schemes are described in Section 8. Two grid quality checks built-in to the grid generation code are described briefly in Section 9. In Section 10, examples from grids produced with the new hyperbolic grid generator for various external components of the Space Shuttle launch vehicle and the SOFIA telescope are presented. Finally, conclusions are given in Section 11.

2. GOVERNING EQUATIONS

Generalized coordinates $\xi(x, y, z)$, $\eta(x, y, z)$, $\zeta(x, y, z)$ are sought where the body surface is chosen to coincide with $\zeta(x, y, z) = 0$ and the surface distributions of $\xi = \text{const}$ and $\eta = \text{const}$ are user-specified. With external aerodynamic applications in mind, the location of the outer boundary $\zeta(x, y, z) = \zeta_{\max}$ is not specified. The governing equations are derived from orthogonality relations between ξ and ζ, between η and ζ, and a cell volume or finite Jacobian J constraint [13]:

$$x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta = 0, \tag{2.1a}$$

$$x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta = 0, \tag{2.1b}$$

$$x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi = \Delta V, \tag{2.1c}$$

or, with $\vec{r}$ defined as $(x, y, z)^T$

$$\vec{r}_\xi \cdot \vec{r}_\zeta = 0, \tag{2.2a}$$

$$\vec{r}_\eta \cdot \vec{r}_\zeta = 0, \tag{2.2b}$$

$$\left| \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \right| = J^{-1} = \Delta V. \tag{2.2c}$$

Equation (2.1) comprise a system of nonlinear partial differential equations in which x, y, and z are specified as initial data at $\zeta = 0$. Local linearization of (2.1) about the state 0 results in the system of grid generation equations

$$A_0(\vec{r} - \vec{r}_0)_\xi + B_0(\vec{r} - \vec{r}_0)_\eta + C_0(\vec{r} - \vec{r}_0)_\zeta = \vec{f} \tag{2.3}$$

with

$$A = \begin{pmatrix} x_\zeta & y_\zeta & z_\zeta \\ 0 & 0 & 0 \\ (y_\eta z_\zeta - y_\zeta z_\eta) & (x_\zeta z_\eta - x_\eta z_\zeta) & (x_\eta y_\zeta - x_\zeta y_\eta) \end{pmatrix}, \tag{2.4a}$$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ x_\zeta & y_\zeta & z_\zeta \\ (y_\zeta z_\xi - y_\xi z_\zeta) & (x_\xi z_\zeta - x_\zeta z_\xi) & (x_\zeta y_\xi - x_\xi y_\zeta) \end{pmatrix}, \tag{2.4b}$$

$$C = \begin{pmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ (y_\xi z_\eta - y_\eta z_\xi) & (x_\eta z_\xi - x_\xi z_\eta) & (x_\xi y_\eta - x_\eta y_\xi) \end{pmatrix}, \tag{2.4c}$$

and

$$\vec{f} = \begin{pmatrix} -\left(\dfrac{\partial \vec{r}}{\partial \xi} \cdot \dfrac{\partial \vec{r}}{\partial \zeta}\right)_0 \\ -\left(\dfrac{\partial \vec{r}}{\partial \eta} \cdot \dfrac{\partial \vec{r}}{\partial \zeta}\right)_0 \\ \Delta V - \Delta V_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \Delta V - \Delta V_0 \end{pmatrix}. \tag{2.4d}$$

We can rewrite (2.3) as

$$A_0 \vec{r}_\xi + B_0 \vec{r}_\eta + C_0 \vec{r}_\zeta + \vec{e} \tag{2.5}$$

with $\vec{e} = (0, 0, \Delta V + 2\Delta V_0)^T$. Now C_0^{-1} exists unless $(\Delta V_0) \to 0$, (2.5) can be rewritten as

$$C_0^{-1} A_0 \vec{r}_\xi + C_0^{-1} B_0 \vec{r}_\eta + \vec{r}_\zeta = C_0^{-1} \vec{e}. \tag{2.6}$$

Although the algebraic verification is not trivial, $C_0^{-1}A_0$ and $C_0^{-1}B_0$ are symmetric matrices [13]. This implies that the linearized system (2.6) is hyperbolic and can be marched with ζ serving as the "time-like" direction.

3. NUMERICAL MARCHING SCHEME

The system of grid generation equations given by (2.5) are solved with a non-iterative implicit finite difference scheme. An unconditionally stable implicit scheme has the advantage that the marching step size in ζ can be arbitrarily selected based only on considerations of accurately generating the grid. Linearization is performed about the previous marching step in ζ.

Let $\Delta\xi = \Delta\eta = \Delta\zeta = 1$ such that $\xi = j - 1$, $\eta = k - 1$, and $\zeta = l - 1$. Central spatial differencing of (2.5) in ξ and η with two-point backward implicit differencing in ζ leads to

$$A_l \delta_\xi \left(\vec{r}_{l+1} - \vec{r}_l\right) + B_l \delta_\eta \left(\vec{r}_{l+1} - \vec{r}_1\right) + C_l \nabla_\zeta \vec{r}_{l-1} = \vec{g}_{l-1}, \tag{3.1}$$

where

$$\vec{g}_{l+1} = \begin{pmatrix} 0 \\ 0 \\ \Delta V_{l+1} \end{pmatrix}$$

and

$$\delta_\xi \vec{r}_j = \frac{\vec{r}_{j+1} - \vec{r}_{j-1}}{2}, \quad \delta_\eta \vec{r}_k = \frac{\vec{r}_{k+1} - \vec{r}_{k-1}}{2},$$

$$\nabla_\zeta \vec{r}_{l+1} = \vec{r}_{l+1} - \vec{r}_l .$$

Throughout, only those indices that change are indicated; thus, $r_{l+1} \Rightarrow r_{j,k,l+1}$ and $r_{j+1} \Rightarrow r_{j+1,k,l}$, etc.

Multiplying through by C_l^{-1} and approximately factoring gives

$$\left(I + C_l^{-1} B_l \delta_\eta\right)\left(I + C_l^{-1} A_l \delta_\xi\right)\left(\vec{r}_{l+1} - \vec{r}_l\right) = C_l^{-1} \vec{g}_{l+1}, \tag{3.2}$$

where I is the identity matrix. The problem is now reduced to solving a sequence of block tridiagonal systems.

Since all ξ- and η-derivatives are approximated by central differencing, numerical dissipation terms are added in these directions. For simplicity, only second differences are used which are explicitly and implicitly included in the basic algorithm as

$$\left[I + C_l^{-1}B_l\delta_\eta - \varepsilon_{i\eta}(\Delta\nabla)_\eta\right]\left[I + C_l^{-1}A_l\delta_\xi - \varepsilon_{i\xi}(\Delta\nabla)_\xi\right]$$

$$\times\left(\vec{r}_{l+1} - \vec{r}_l\right) = C_l^{-1}\vec{g}_{l+1} - \left[\varepsilon_{e\xi}(\Delta\nabla)_\xi + \varepsilon_{e\eta}(\Delta\nabla)_\eta\right]\vec{r}_l, \quad (3.3)$$

where, for example,

$$(\Delta\nabla)_\eta\vec{r} = \vec{r}_{k+1} - 2\vec{r}_k + \vec{r}_{k-1},$$

and with $\varepsilon_{i\xi} \approx 2\varepsilon_{e\xi}$ and $\varepsilon_{i\eta} \approx 2\varepsilon_{e\eta}$. Additional smoothing and implicitness [5] are achieved by differencing $\nabla_\zeta\vec{r} = \vec{F}$ as $\vec{r}_{l+1} - \vec{r}_l = (1+\theta)\vec{F}_{l+1} - \theta\vec{F}_l$, with $\theta \geqslant 0$. This differencing in ζ is incorporated into (3.3) as

$$\left[I + (1+\theta_\eta)C_l^{-1}B_l\delta_\eta - \varepsilon_{i\eta}(\Delta\nabla)_\eta\right]$$

$$\times\left[I + (1+\theta_\xi)C_l^{-1}A_l\delta_\xi - \varepsilon_{i\xi}(\Delta\nabla)_\xi\right]$$

$$\times\left(\vec{r}_{l+1} - \vec{r}_l\right) = C_l^{-1}\vec{g}_{l+1} - \left[\varepsilon_{e\xi}(\Delta\nabla)_\xi + \varepsilon_{e\eta}(\Delta\nabla)_\eta\right]\vec{r}_l. \quad (3.4)$$

The values of θ_ξ and θ_η are kept at zero unless the body contains concave profiles in ξ or η, in which case values of the appropriate θ of 1–4 are effective in preventing grid lines from crossing.

The coefficient matrices A_l, B_l and C_l contain derivatives in ξ, η and ζ. The derivatives in ξ and η are obtained by central differencing while the derivatives in ζ are obtained from (2.1) as a linear combination of ξ- and η-derivatives as follows

$$\begin{pmatrix} x_\zeta \\ y_\zeta \\ z_\zeta \end{pmatrix} = \frac{\Delta V}{\mathrm{Det}(C)}\begin{pmatrix} y_\xi z_\eta - y_\eta z_\xi \\ x_\eta z_\xi - x_\xi z_\eta \\ x_\xi y_\eta - x_\eta y_\xi \end{pmatrix} = C^{-1}\vec{g} \quad (3.5)$$

with

$$\mathrm{Det}(C) = (y_\xi z_\eta - y_\eta z_\xi)^2 + (x_\eta z_\xi - x_\xi z_\eta)^2 + (x_\xi y_\eta - x_\eta y_\xi)^2.$$

A discussion on a more appropriate way to compute these ζ-derivatives is given in Section 7.

4. BOUNDARY CONDITIONS

Five types of implicit boundary conditions have been implemented in the grid generation code at the ξ and η boundaries (except for the axis condition, which is only implemented in ξ). In the following, coordinate increments $\Delta \vec{r} = \vec{r}_{l+1} - \vec{r}_l$ are represented by $(\Delta x, \Delta y, \Delta z)^T$.

1. *Periodicity*: All derivatives at the end points in the periodic direction are evaluated by 'wrapping around.' A periodic block-tridiagonal solver is used for the inversion of the appropriate factor in the left-hand side of (3.4).

2. *Constant Cartesian plane*: If a ξ or η boundary is restricted to an x = const, y = const or z = const plane, then that value is enforced and the other variables are 'floated.' For example, for an x = const plane at the $j = 1$ boundary, x is held constant and y and z are floated using

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}_{j=1} = \begin{pmatrix} 0 \\ \Delta y \\ \Delta z \end{pmatrix}_{j=2}. \qquad (4.1)$$

3. *Symmetry plane*: Conventional reflection planes are used to impose symmetry about any $x = 0$, $y = 0$ or $z = 0$ plane and values are updated implicitly. For example, to update a reflected plane at $j = 1$ for a symmetry condition about $x = 0$ corresponding to $j = 2$, x reflects odd and y and z reflect even as:

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \end{pmatrix}_{j=1} = \begin{pmatrix} -\Delta x \\ \Delta y \\ \Delta z \end{pmatrix}_{j=3}. \qquad (4.2)$$

Since the $j = 2$ plane in this example may deviate very slightly from $x = 0$ away from the body surface due to round-off error, post-processing is done to set the x coordinate exactly to zero at $j = 2$ at each incremental level in l.

4. *Floating edge*: Much as in the case of a constant Cartesian plane, an entire ξ or η boundary can be floated using the simple hyperbolic equation $\vec{r}_{\xi\zeta} = 0$ or $\vec{r}_{\eta\zeta} = 0$ to update a boundary plane. This is essentially a zeroth-order extrapolation of $\Delta\vec{r}$ from the adjacent interior value, and it often works well. Occasionally, however, the floating boundary plane itself may tend to 'roll in' or 'kink' while its neighboring planes in the interior remain smooth. Since the interior points are fine, this problem has been remedied by the addition of a fictitious line of points on the body surface next to the floating edge by linear extrapolation from the interior. The dimension of the surface grid in a particular direction is temporarily increased by one or two depending on whether one or two floating edges are present in that direction. The 3D grid is then generated over the extended surface grid, after which the fictitious plane or planes of points is removed. The addition and removal of these fictitious points are carried out internal to the code and is not a concern for the user.

When using the chimera overset-grid scheme, it is particularly desirable to have the floating edges splay outwards, i.e., in the direction away from the interior of the grid, thus providing better overlap between neighboring grids. Although a free floating edge using zeroth-order extrapolation may bend inwards or outwards, it is found that using a mixed zeroth- and first-order extrapolation scheme tends to bend the edge outwards. For example, at the $j = 1$ boundary, we have

$$\begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix}_{j=1} = \begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix}_{j=2} + \varepsilon_{ex}\left[\begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix}_{j=2} - \begin{pmatrix}\Delta x\\ \Delta y\\ \Delta z\end{pmatrix}_{j=3}\right], \qquad (4.3)$$

where $0 \leqslant \varepsilon_{ex} \leqslant 1$. Zeroth-order and first-order extrapolation schemes are recovered at the two limits of ε_{ex} respectively. Typically, a value of $\varepsilon_{ex} = 0.05-0.2$ is used. An example using the floating edge condition with $\varepsilon_{ex} = 0.2$ at the boundaries of a flat plate is shown in Figure 1.

5. *Axis*: When the axis logic is used in the j-direction, it is assumed that the boundary condition in the k-direction is either periodic, or that of symmetry, or constant planes at both ends. For example, one may generate a grid for a complete ellipsoid (periodic in k) or just one half of it (symmetry or constant planes at end points in k).

The treatment of the axis requires special attention in order to produce smooth results. Two methods have been developed. The first method requires the user to adjust certain input parameters and is able to produce smooth results for all cases encountered so far. The second method solves the

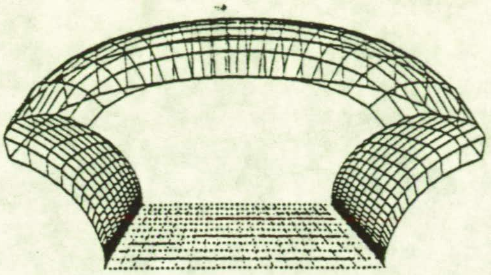


FIG. 1. Outward-splaying edges of a flat plate.

governing equations on a local coordinate system spanning points to either side of the axis. It has the advantage of having fewer parameters to adjust, but it is not appropriate for a conical-like axis and will not be described here.

The first method involves using a mixed zeroth- and first-order extrapolation and volume scaling. The axis point is updated implicitly by imposing that $\Delta \vec{r}$ at the axis is extrapolated by a mixed zeroth- and first-order scheme similar to that given by (4.3). The resulting $k_{\max}$ predicted values, where $k_{\max}$ is the number of points around the axis, are averaged to produce a unique value at the axis. The methods of volume specification described in Section 5 below usually produce volumes that are too large near the axis. Hence, the volumes in the ring of points around the axis are scaled by a reduction factor in the range 0.1–1.0. Typical values of the extrapolation factor and volume scaling factor are 0.4 and 0.3, respectively. Figure 2 shows the symmetry plane of an external tank grid with an axis at the pointed nose and an axis at the flat back. Smoothness is maintained at both the front and the back regions by application of the above scheme.

5. CELL VOLUME SPECIFICATION

With the hyperbolic grid generation method, one of the means of controlling the grid is by specification of the cell volumes, $\Delta V_{j,k,l}$. Through the cell volumes, the extent and clustering of the grid can be modified. Since the cell volume at each point must be given, it is clear that the user must devise a simple global method for specifying volumes.

One method is that the specified volume at each point is set equal to the computed surface area element times a user-specified arc length. Specifically,

$$\Delta V_{j,k,l} = \Delta s_{j,k,l} \, \Delta A_{j,k,l}, \tag{5.1}$$

ORIGINAL PAGE IS
OF POOR QUALITY

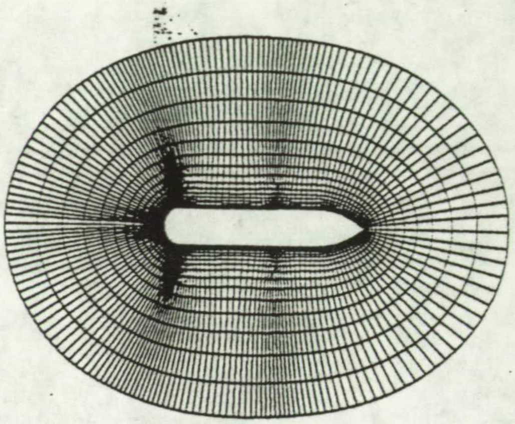


FIG. 2. Symmetry plane of external tank grid with a pointed axis in front and a flat axis at the back.

where $\Delta s_{j,k,l} = s_{j,k,l+1} - s_{j,k,l}$ is the user-specified arc length for marching and $\Delta A_{j,k,l}$ is the surface area element. The radial point distribution function $s_{j,k,l}$ prescribes the arc length between points in the direction normal to the body surface. In the most general case, s is a function of j, k and l since each point on the body surface may be stretched to a different outer boundary location. (For example, in the case of a body at a positive angle of attack in hypersonic flow, one may wish to have the outer boundary of the grid farther away on the top surface than on the lower surface). Typically, the points are stretched away from the body exponentially. If grid point spacing control is required at the outer boundary as well as at the body surface, or if more uniform grid point spacing is required away from the inner body surface, hyperbolic tangent stretching can be used. The ability to control the grid spacing at each end of the domain is useful when multiple zones consisting of stretched points or uniformly-spaced points are desired. Also, the presence of more points in the far field of a component grid helps to improve grid-overlap regions for the chimera overset-grid scheme [2].

In this kind of volume control specification, if an initial distribution of points is highly clustered in ξ or η, then these points tend to remain highly clustered even far away from the body. In order to obtain a more uniform far-field distribution, the volumes specified from (5.1) are averaged in ξ and η with each step taken in ζ. For example, the averaged volume $\Delta \bar{V}_{j,k,l}$ can be computed as

$$\Delta \bar{V}_{j,k,l} = (1 - \nu_a)\, \Delta V_{j,k,l} + \frac{\nu_a}{4} \times \left(\Delta V_{j+1,k,l} + \Delta V_{j-1,k,l} + \Delta V_{j,k+1,l} + \Delta V_{j,k-1,l} \right), \quad (5.2)$$

where this is applied one or more times with each step in ζ. A typical value of ν_a that has been employed is 0.16.

An alternative method for specifying mesh cell volumes is to form a grid about a 'similar' but simple reference body for which the grid can be generated analytically, and to use appropriately scaled cell volumes from this reference grid for the more complex problem. This method is described in more detail in [13].

6. SMOOTHING

Since central differencing is used in the numerical scheme, artificial dissipation terms are added to control oscillations arising from the odd-even uncoupling of grid points. An equally important function of the added dissipation is to control the smoothness of the resulting grid. The form and magnitude of the added dissipation are very important in shaping the grid quality.

The parameters θ_ξ and θ_η on the left-hand side of the grid generation Equation (3.4) can be thought of as a type of smoothing in the ζ marching direction. In the ξ and η directions, it is adequate for the purpose of controlling the smoothness of the grid to use just second-order smoothing. The form of this smoothing is described below.

For simple surface topologies, such as an ellipsoid that has a convex profile in all directions, a constant dissipation coefficient is sufficient to provide good grid quality. However, body surfaces encountered in aerodynamic applications are frequently much more complex with combinations of sharp convex and concave corners. Hence, the dissipation, which tends to reduce grid curvature and orthogonality, must be applied more selectively. The reduction of grid curvature can be advantageous in concave regions and detrimental in convex regions. In order to prevent grid lines from converging and crossing, the dissipation has to be relatively high in concave regions (e.g., in the region between the Orbiter fuselage and the wing root in Figure 3). Conversely, the dissipation must be kept small near the body surface and sharp convex corners (such as the convex corners of the IEA box in Figure 4), otherwise the resulting grid spacing in the ζ direction would become too reduced or even negative. A spatially uniform dissipation coefficient is unable to satisfy all of the above requirements. Spatially-varying dissipation coefficients have been used previously, but they have only accounted for variations of mesh size, specifically, $C^{-1}A$ and $C^{-1}B$ variations. A spatially-varying form of the dissipation coefficient that has worked very well for many cases is described below.

The explicit second-order dissipation D_e added to the right hand side of

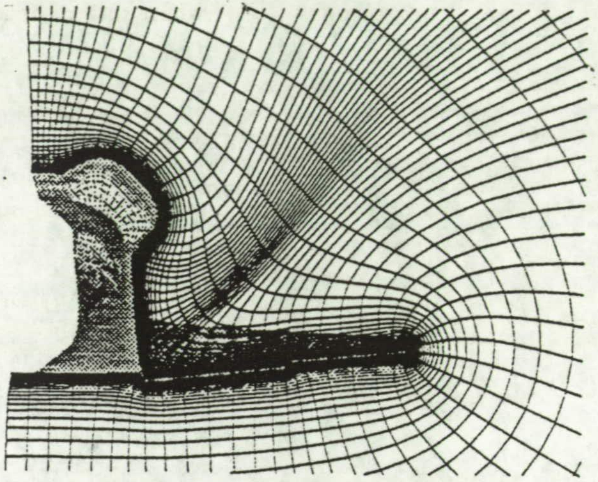


FIG. 3. Plane on the back section of the Orbiter.

the equations is given by

$$D_e = -\left[\varepsilon_{e\xi}(\Delta\nabla)_\xi + \varepsilon_{e\eta}(\Delta\nabla)_\eta\right]\vec{r}_l, \tag{6.1}$$

with

$$\varepsilon_{e\xi} = \varepsilon_e R_\xi N_\xi, \quad \varepsilon_{e\eta} = \varepsilon_e R_\eta N_\eta, \tag{6.2}$$

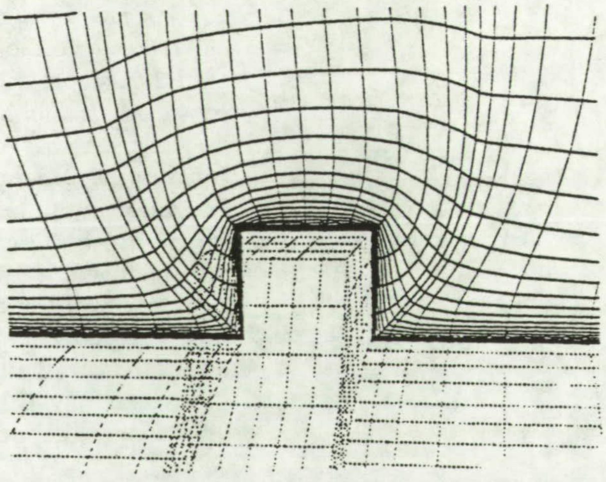


FIG. 4. Plane through IEA box on the ring of the solid rocket booster.

where ε_c is a user-specified constant of $O(1)$. N_ξ and N_η are approximations to the matrix norms $\|C^{-1}A\|$ and $\|C^{-1}B\|$, respectively, given by

$$N_\xi = \sqrt{\frac{x_\zeta^2 + y_\zeta^2 + z_\zeta^2}{x_\xi^2 + y_\xi^2 + z_\xi^2}}, \quad N_\eta = \sqrt{\frac{x_\zeta^2 + y_\zeta^2 + z_\zeta^2}{x_\eta^2 + y_\eta^2 + z_\eta^2}}, \tag{6.3}$$

and R_ξ, R_η are the dissipation coefficients given by

$$R_\xi = S_l \bar{d}^\xi_{j,k,l} a^\xi_{j,k,l}, \; R_\eta = S_l \bar{d}^\eta_{j,k,l} a^\eta_{j,k,l}. \tag{6.4}$$

The dissipation coefficients consist of three functions that provide different ways to automatically adjust the local dissipation appropriately, depending on the local grid topology:

(1) A scaling function S_l, which varies with normal distance from the body surface.

(2) A grid point distribution sensor function, $\bar{d}^\xi_{j,k,l}$ or $\bar{d}^\eta_{j,k,l}$, depending on the direction, which senses mesh convergence based on the distances between neighboring grid points.

(3) A grid angle function, $a^\xi_{j,k,l}$ or $a^\eta_{j,k,l}$, depending on the direction, which senses mesh convergence based on the angles between neighboring grid points.

The form of the scaling function S_l is given by

$$S_l = \begin{cases} \sqrt{\dfrac{(l-1)}{(l_{\max}-1)}} & 2 \leqslant l \leqslant l_{\text{trans}}, \\[2ex] \sqrt{\dfrac{(l_{\text{trans}}-1)}{(l_{\max}-1)}} & l_{\text{trans}} + 1 \leqslant l \leqslant l_{\max}. \end{cases} \tag{6.5}$$

where $l_{\max}$ is the number of points in the l direction and l_{trans} is restricted to the range $[\frac{3}{4}, 1] \times l_{\max}$. With $d^\xi_{j,k,l}$ and $d^\eta_{j,k,l}$ defined by (6.8a, b), l_{trans} is set to l when one or both of the following is true:

$$\max_{j,k} d^\xi_{j,k,l} - \max_{j,k} d^\xi_{j,k,l-1} < 0, \tag{6.6a}$$

$$\max_{j,k} d^\eta_{j,k,l} - \max_{j,k} d^\eta_{j,k,l-1} < 0. \tag{6.6b}$$

Once l_{trans} is located, the above tests are not performed for $l > l_{\text{trans}}$.

The purpose of the scaling function S_l is to guarantee small dissipation, and hence, grid orthogonality near the body surface. As one moves away from the body surface, dissipation is increased since grid lines may begin to converge in concave regions and some loss of grid curvature in convex regions is no longer a problem. Away from the body surface, the grid point distribution sensor function and the grid angle function alone are sufficient to provide the appropriate amount of dissipation. Hence, the influence of S_l is removed by setting it to a constant at some location l_{trans} away from the body. It is found that a good criterion for locating l_{trans} is when the convergence of local grid lines is slowing down in some sense given by (6.6a, b). For all the cases encountered so far, it is sufficient to test for l_{trans} for $l \geqslant \frac{3}{4} l_{\max}$. For some cases, l_{trans} may be less than $l_{\max}$, which reduces loss of orthogonality near the outer boundary; while for other cases, l_{trans} may have to be equal to $l_{\max}$, when increasing values of dissipation are needed all the way to the outer boundary to prevent grid lines from converging.

The forms of the grid point distribution sensor functions $\tilde{d}^{\xi}_{j,k,l}$, $\tilde{d}^{\eta}_{j,k,l}$ are given by

$$\begin{aligned} \tilde{d}^{\xi}_{j,k,l} &= \max\left[\left(d^{\xi}_{j,k,l}\right)^{2/S_l}, 0.1\right], \\ \tilde{d}^{\eta}_{j,k,l} &= \max\left[\left(d^{\eta}_{j,k,l}\right)^{2/S_l}, 0.1\right], \end{aligned} \tag{6.7}$$

where

$$d^{\xi}_{j,k,l} = \frac{|\bar{r}_{j+1,k,l-1} - \bar{r}_{j,k,l-1}| + |\bar{r}_{j-1,k,l-1} - \bar{r}_{j,k,l-1}|}{|\bar{r}_{j+1,k,l} - \bar{r}_{j,k,l}| + |\bar{r}_{j-1,k,l} - \bar{r}_{j,k,l}|}, \tag{6.8a}$$

$$d^{\eta}_{j,k,l} = \frac{|\bar{r}_{j,k+1,l-1} - \bar{r}_{j,k,l-1}| + |\bar{r}_{j,k-1,l-1} - \bar{r}_{j,k,l-1}|}{|\bar{r}_{j,k+1,l} - \bar{r}_{j,k,l}| + |\bar{r}_{j,k-1,l} - \bar{r}_{j,k,l}|}. \tag{6.8b}$$

The distribution of grid points in the ξ and η directions are monitored by the functions $d^{\xi}_{j,k,l}$ and $d^{\eta}_{j,k,l}$, respectively. The quantity $d^{\xi}_{j,k,l}$ is the ratio of the distances between a grid point and its neighbors in the ξ direction at level $(l-1)$ to that at level l. This ratio is high in concave regions and hence, more dissipation is provided here. It is of order one or smaller in flat or convex regions, where less dissipation is needed. Similarly, $d^{\eta}_{j,k,l}$ represents the corresponding quantity in the η direction. The quantities $\tilde{d}^{\xi}_{j,k,l}$ and $\tilde{d}^{\eta}_{j,k,l}$ are constructed from $d^{\xi}_{j,k,l}$ and $d^{\eta}_{j,k,l}$, respectively, which are raised to the power $2/S_l$ in order to counteract the small value of S_l close to the body surface. Also, the values of $\bar{d}^{\xi}_{j,k,l}$ and $\bar{d}^{\eta}_{j,k,l}$ are limited from becoming too low by a limiter of 0.1. Note that a grid point distribution sensor function based on cell area ratios is not as effective since the grid lines could be converging in one direction but not the other and the cell areas would not decrease very much.

The grid angle functions $a^{\xi}_{j,k,l}, a^{\eta}_{j,k,l}$ are more conveniently defined in terms of the following unit vectors. Let the vectors pointing in the plus and minus ξ directions at grid point (j, k, l) be represented by $\vec{r}_j^{\,+}$ and $\vec{r}_j^{\,-}$, respectively, where

$$\vec{r}_j^{\,+} = \vec{r}_{j+1,k,l} - \vec{r}_{j,k,l}, \quad \vec{r}_j^{\,-} = \vec{r}_{j-1,k,l} - \vec{r}_{j,k,l}, \tag{6.9}$$

and let $\hat{r}_j^+$ and $\hat{r}_j^-$ be the respective unit vectors for $\vec{r}_j^{\,+}$ and $\vec{r}_j^{\,-}$. Similar expressions are defined for $\vec{r}_k^{\,+}$, $\vec{r}_k^{\,-}$ and $\hat{r}_k^+$, $\hat{r}_k^-$ for vectors and unit vectors pointing in the plus and minus η directions, respectively at grid point (j, k, l). The local unit normal $\hat{n}_{j,k,l}$ based on the cross product of the above unit vectors is given by

$$\hat{n}_{j,k,l} = \frac{(\hat{r}_j^+ - \hat{r}_j^-) \times (\hat{r}_k^+ - \hat{r}_k^-)}{\left|(\hat{r}_j^+ - \hat{r}_j^-) \times (\hat{r}_k^+ - \hat{r}_k^-)\right|}. \tag{6.10}$$

The cosine of the local half angle $\alpha_{j,k,l}$ in the ξ direction is then given by

$$\cos \alpha_{j,k,l} = \hat{n}_{j,k,l} \cdot \hat{r}_j^+ = \hat{n}_{j,k,l} \cdot \hat{r}_j^- . \tag{6.11}$$

The grid angle function $a^{\xi}_{j,k,l}$ is then defined as

$$a^{\xi}_{j,k,l} = \begin{cases} 1/\left(1 - \cos^2 \alpha_{j,k,l}\right) & \text{if } 0 \leqslant \alpha_{j,k,l} \leqslant \dfrac{\pi}{2}, \\ 1 & \text{if } \dfrac{\pi}{2} < \alpha_{j,k,l} \leqslant \pi. \end{cases} \tag{6.12}$$

A similar expression is defined for the grid angle function $a^{\eta}_{j,k,l}$ in terms of the cosine of the local half angle $\beta_{j,k,l}$ in the η direction.

The smoothing provided by the grid point distribution functions in (6.7) has to be modified locally at grid points located at very sharp concave corners. These corners are detected by computing the half angles α and β subtended by neighboring grid points in the ξ and η directions respectively (see (6.11)). At a severe concave corner point, extra dissipation is required to prevent crossing of grid lines. The functions $a^{\xi}_{j,k,l}$ and $a^{\eta}_{j,k,l}$ serve to provide the appropriate local modifications to the dissipation required at these corner points. For a typical mesh, the values of $a^{\xi}_{j,k,l}$ and $a^{\eta}_{j,k,l}$ are close to unity at most grid points. Figure 5 shows the grid marching out of a concave angle of 20 degrees. Note that orthogonality is still maintained for the first point off the surface. Grids for concave angles down to 5 degrees have been obtained with the above algorithm.

A minor local refinement at $l = 2$ of the above dissipation scheme for sharp corners is given in the Appendix.

7. METRIC CORRECTION

Often a body surface will have a corner or edge region that is a discontinuity to the hyperbolic grid equations. Nevertheless, the spatially-varying dissipation described in Section 6 will typically generate a sufficiently smooth grid at either convex or concave corners if the grid spacing to each side of the corner is approximately equal. If the surface grid possesses corners with uneven grid spacings, dissipation alone is usually not sufficient to provide a smooth volume grid and additional remedies are needed. The metric correction procedure described below generally provides a satisfactory treatment of corner discontinuities in all but extremely convex cases.

With ζ-derivatives as defined in (3.5), the direction in which the grid will emanate from the corner is such that it is perpendicular to the line joining the two neighbor points of the corner. For a corner with unequally spaced points, this is clearly not a desirable feature (see Figure 6a). In order to guide the grid out in a direction that bisects the angles at a point subtended by its neighbors in both the ξ and η directions, the derivatives $x_\zeta, y_\zeta, z_\zeta$ at the point are modified to $x'_\zeta, y'_\zeta, z'_\zeta$ as follows:

$$\begin{pmatrix} x'_\zeta \\ y'_\zeta \\ z'_\zeta \end{pmatrix} = \frac{\Delta V}{Det(C')} \begin{pmatrix} y'_\xi z'_\eta - y'_\eta z'_\xi \\ x'_\eta z'_\xi - x'_\xi z'_\eta \\ x'_\xi y'_\eta - x'_\eta y'_\xi \end{pmatrix}. \tag{7.1}$$

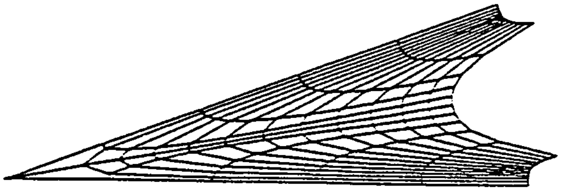


FIG. 5. A concave corner at 20 degrees.

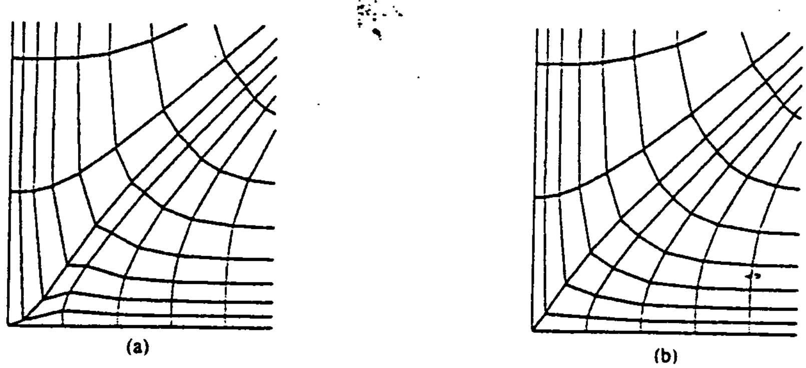


FIG. 6. Comparison of treatment of concave corner with unequal grid spacings. (a) Without metric correction, (b) with metric correction.

with

$$Det(C') = (y'_\xi z'_\eta - y'_\eta z'_\xi)^2 + (x'_\eta z'_\xi - x'_\xi z'_\eta)^2 + (x'_\xi y'_\eta - x'_\eta y'_\xi)^2,$$

where

$$(x'_\xi, y'_\xi, z'_\xi)^T = \frac{1}{4}\left(|\vec{r}_j^{\,+}| + |\vec{r}_j^{\,-}|\right)\left(\frac{\vec{r}_j^{\,+}}{|\vec{r}_j^{\,+}|} - \frac{\vec{r}_j^{\,-}}{|\vec{r}_j^{\,-}|}\right), \qquad (7.2a)$$

and

$$(x'_\eta, y'_\eta, z'_\eta)^T = \frac{1}{4}\left(|\vec{r}_k^{\,+}| + |\vec{r}_k^{\,-}|\right)\left(\frac{\vec{r}_k^{\,+}}{|\vec{r}_k^{\,+}|} - \frac{\vec{r}_k^{\,-}}{|\vec{r}_k^{\,-}|}\right). \qquad (7.2b)$$

While the ζ-derivatives should be computed by (7.1) near the body surface, the original method of computing these quantities should be restored away from the body surface. This can be achieved smoothly by

$$(x_\zeta, y_\zeta, z_\zeta)^T = (1 - \nu_l)(x''_\zeta, y''_\zeta, z''_\zeta)^T + \nu_l(x'_\zeta, y'_\zeta, z'_\zeta)^T, \qquad (7.3)$$

where $\nu_l = 2^{2-l}$ and $x''_\zeta, y''_\zeta, z''_\zeta$ are computed by (3.5). The form of (7.1) is the same as (3.5) except that the ξ- and η-derivatives are replaced by the corresponding primed quantities. These modified ξ- and η-derivatives are constructed in such a way that the neighboring points of the corner appear to be of equal distance from the corner. The result of applying this procedure to a concave corner is shown in Figure 6b.

8. TREATMENT OF CONVEX CORNERS

Two methods are presented here that provide extra robustness at convex corners. They both involve switching from solving the hyperbolic grid generation Equation (3.4) to some other equations at the convex corner point. The first method is an implicit averaging scheme. Instead of solving the hyperbolic equations at a convex corner, the following average equation is solved.

$$\Delta \vec{r}_{j,k} = \tfrac{1}{2}(\mu_\xi + \mu_\eta)\Delta \vec{r}_{j,k}, \tag{8.1}$$

where

$$\mu_\xi \Delta \vec{r}_{j,k} = \tfrac{1}{2}\left(\Delta \vec{r}_{j+1,k} + \Delta \vec{r}_{j-1,k}\right), \tag{8.2a}$$

$$\mu_\eta \Delta \vec{r}_{j,k} = \tfrac{1}{2}\left(\Delta \vec{r}_{j,k+1} + \Delta \vec{r}_{j,k-1}\right). \tag{8.2b}$$

In other words, the marching increment at the corner is the average of the marching increment of its four neighboring points. The form of the above scheme can be made compatible with the hyperbolic grid generation scheme given by (3.4) by approximate factorization. The equation to be solved at convex corner points is then

$$\left(I - \tfrac{1}{2}\mu_\xi\right)\left(I - \tfrac{1}{2}\mu_\eta\right)\Delta \vec{r} = 0. \tag{8.3}$$

At $l = 2$, the switch to solve (8.3) is performed if the external angle of a convex corner in *either* the ξ or η direction exceeds 240 degrees. As the grid marches out in l, the switch to return to the normal hyperbolic scheme is performed when the minimum of $\cos \alpha_{j,k}$ and $\cos \beta_{j,k}$ becomes larger than about -0.2. Although some factorization error is present, this implicit averaging scheme has worked well for a variety of convex corners, including some cases where the normal scheme has failed. One such case is illustrated by the NACA0012 airfoil shown in Figures 7a and 7b. The point of this example is *not* to show that an O-grid should be used for the airfoil but simply to demonstrate the ability of the scheme to produce high quality grids around sharp convex corners.

The implicit averaging scheme described above possesses two desirable properties. It can be shown geometrically that if the grid marches out

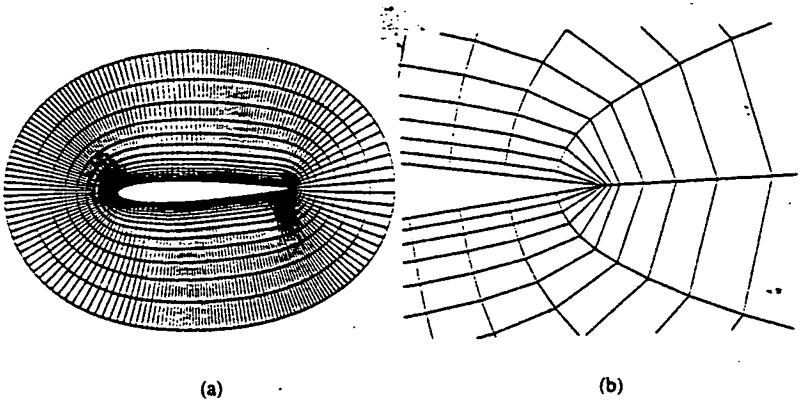


FIG. 7. Airfoil with sharp trailing edge and O-grid topology. (a) Far view, (b) close up view of sharp trailing edge region.

orthogonally at the neighbors of the convex corner, then the distance marched out at the corner point is always smaller than the distance marched out at the neighboring points. Moreover, the distance marched out at the corner point becomes smaller as the convex corner becomes sharper. This is very useful in helping to bend the neighboring grid lines towards the sharp corner as the grid is marched out away from the body surface (see Figure 7b). The second property is that the grid marching out from the corner point will bisect the angle at the corner provided the neighboring points march out the same distance and in symmetrical directions to each other.

An alternative but potentially more robust method than the implicit averaging scheme above is described below. The exact location of the grid point in the next marching step out from a convex corner is predicted in advance. The predicted point is located by marching the grid a distance of $\Delta \bar{s}_{j,k,l}$ in the direction given by the angle-bisecting unit normal defined by $(x'_\zeta, y'_\zeta, z'_\zeta)^T$ in (7.1). The marching distance is scaled as

$$\Delta \bar{s}_{j,k,l} = \Delta s_{j,k,l} \times \min(\sin \alpha_{j,k,l}, \sin \beta_{j,k,l}), \tag{8.4}$$

where $\Delta s_{j,k,l}$ is the user-specified arc length in the normal direction (see Section 5). The scaling with the sine of the half angle causes the corner point to march out a smaller distance than its neighbors, thus helping to bend the neighboring grid lines towards the corner. Thus, $\Delta \bar{r}$ at the convex corner point can be computed in advance and combined with the grid generation Equation (3.4). From numerical trials with different geometries, it was found that this procedure also works well for many types of convex corners.

9. GRID QUALITY CHECKS

Two grid quality checks are performed in the hyperbolic grid generation code. The first check is a cell volume check by decomposing each cell into six tetrahedra [7]. This test is only passed if the volume of each tetrahedron in the cell is positive. The second check involves computing the Jacobian of each cell using a finite volume scheme employed by typical flow solvers. Negative volumes or Jacobians are reported to the user and smoothing parameters can then be adjusted appropriately to remove the bad cells.

10. RESULTS AND APPLICATIONS

The vectorized version of the current hyperbolic grid generator runs at 142 megaflops on the CRAY-YMP and requires about 9.7 microseconds of CPU time per grid point (i.e., about a hundred thousand points per CPU second). As an example, generation of the largest grid in the Space Shuttle launch vehicle grid system, the Orbiter grid for flight Reynolds number (98 × 77 × 57 = 430122 points), takes 4.17 seconds of CPU time. This is about one to two orders of magnitude faster than typical elliptic grid generators.

The various external components of the integrated Space Shuttle vehicle contain a wide variety of different geometric features that are found in many other applications. Hence, these geometric components provide good tests of robustness for the hyperbolic grid generation scheme described above. Some examples are given below.

The following two examples show how the spatially-variable dissipation coefficient works at different types of corners. The first example shows the grid around a sequence of sharp convex and concave corners that appears at the IEA box on the attach ring of the solid rocket booster as in Figure 4. Low dissipation values are needed at the body surface and above the convex corners to maintain orthogonality while high dissipation values are needed in the concave regions to provide grid smoothness. Figure 3 shows the smoothness of the grid in a large concave region, the wing root region at the back of the Orbiter.

The robustness of the axis logic is demonstrated in the next example where the geometry is noncircular in the circumferential direction around the axis and that the distribution of grid points in this direction is nonuniform. This type of axis condition is present at the tip of the vertical tail section of the Orbiter as in Figure 8. The mixed extrapolation scheme with volume scaling is able to produce a smooth grid for this case.

When the chimera overset-grid method is used on two grids whose body surfaces intersect each other, the grid points in the region around the

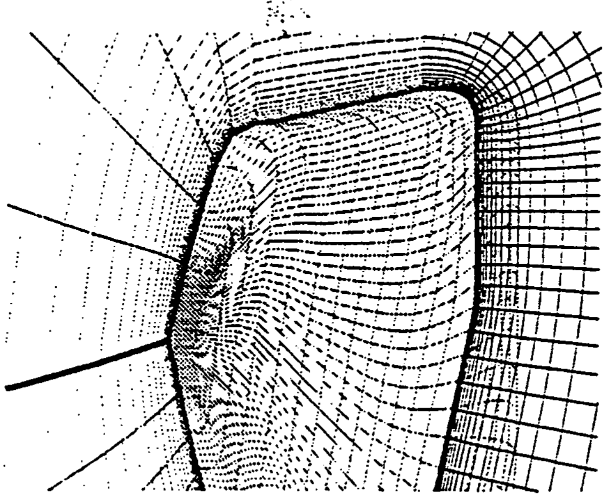
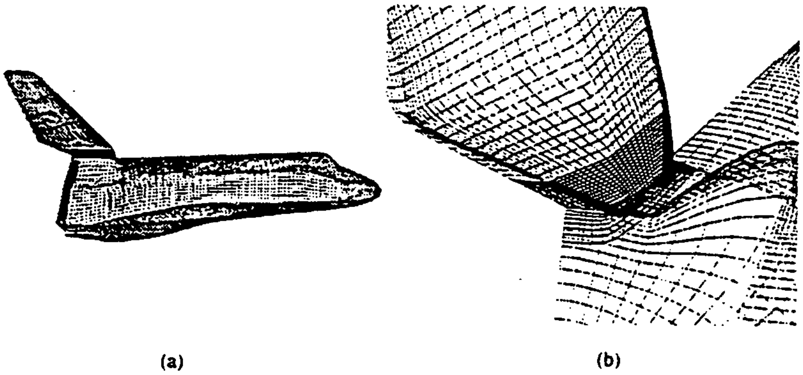


FIG. 8. Vertical tail grid showing part of the surface grid near the tip, the 3D grid on the symmetry plane and a section normal to the symmetry plane. The axis is marked by the thick dark line.

intersection line which are common to both grids are left with no interpolation stencils. In order to remedy this problem, a collar grid can be introduced which covers the region around the intersection line [11]. The most challenging example tested by the grid generator so far is the collar grid, which covers the intersection region between the vertical tail and the Orbiter as in Figures 9a, b. The methods used to generate the collar grid surface are explained in [9]. The surface of the collar grid is made up of two parts. The top part lies on



(a) (b)

FIG. 9. Views of the collar grid surface. (a) Position relative to vertical tail and Orbiter. (b) close up view relative to vertical tail and Orbiter.

the surface of the tail down to the intersection line with the Orbiter. The lower part has two sections. The first section starts at the intersection line with the tail and then follows the top surface of the Orbiter. The second section folds over the back of the Orbiter and follows the backward-facing aft-bulkhead of the Orbiter. The difficult feature of this geometry is the presence of a region where grid lines are concave in one direction and convex in the other. Slices of the 3D grid viewed from the front and back ends of the collar grid are shown in Figures 10a, b.

The final example is taken from the telescope grid for the SOFIA vehicle [1]. The SOFIA is a modified Boeing 747 with a telescope mounted inside a cavity on the upper surface of the plane. The telescope is topologically similar to a hollow bowl with a truncated cylinder in the middle of the inside of the bowl. Figure 11a shows the surface geometry (shaded) for half the telescope and slices of the 3D grid. The external surface of the telescope consists of both the outside and inside of the bowl together with the middle cylinder. The symmetry plane of the 3D grid is shown in Figure 11b. These figures show that, although the surface grid may not possess the sufficient number of points to resolve the detailed flow structures, the grid generator is able to produce a smooth grid over the complex combination of concave and convex corners using the techniques described in Sections 6, 7 and 8. The outer boundary of the telescope grid need not be placed far away from the body surface since the entire telescope grid is surrounded by a larger cavity grid and communication between the two grids is achieved via the chimera overset-grid scheme.

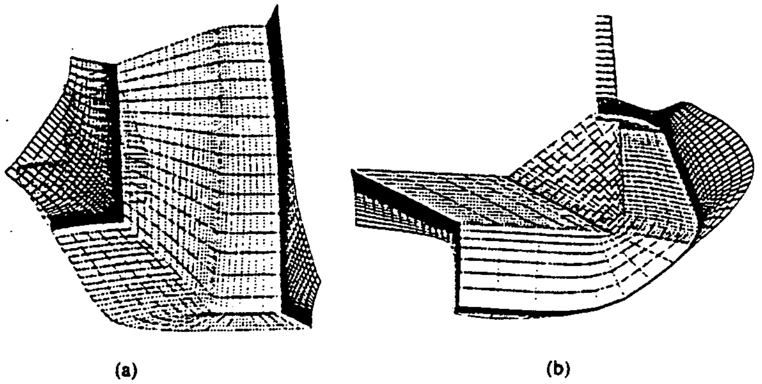

(a) (b)

FIG. 10. Views of sections of the 3D collar grid joining the vertical tail and the Orbiter. (a) Front view, (b) back view.

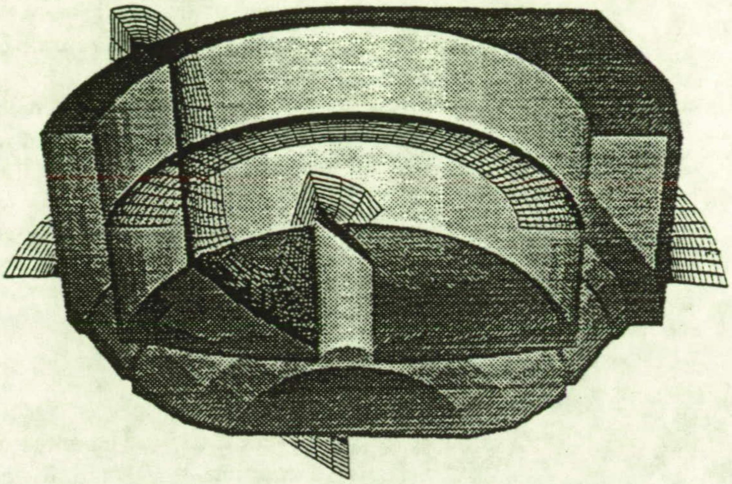


FIG. 11a. Surface geometry and slices of the 3D grid for the SOFIA telescope.

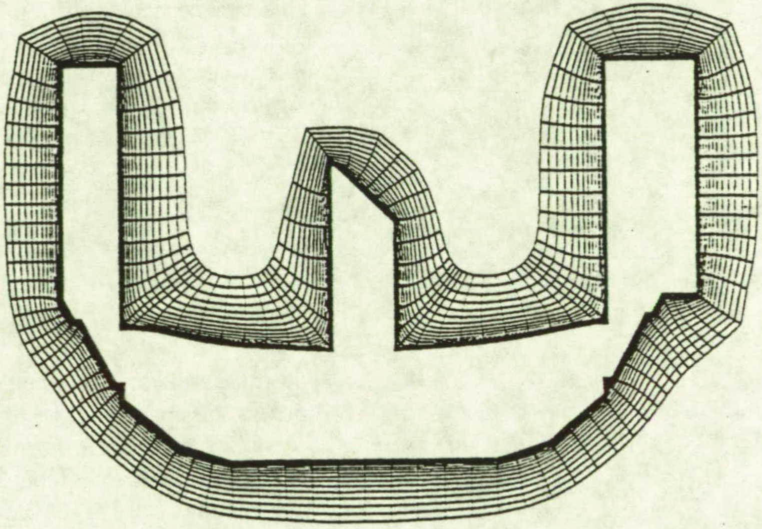


FIG. 11b. Symmetry plane of the 3D grid for the SOFIA telescope.

11. CONCLUSIONS

A robust three-dimensional hyperbolic grid generation scheme that is able to produce high quality grids for a wide variety of geometries has been presented. The improved robustness and the speed advantage of the scheme have made it extremely attractive for users of chimera flow-solvers. Since the scheme is fast, the user can readily adjust the input parameters to fine-tune the grid quality.

The use of a spatially-varying dissipation coefficient based on distances and angles between neighboring grid points gives the grid generator the ability to cope with geometries that are more complex than before. The grid angle bisecting property is provided by the metric correction procedure near the body surface. Sharp convex corners are automatically detected and the grid generation equations are altered at these corners to further enhance smoothness and robustness.

APPENDIX

In order to guarantee orthogonality near the body surface, the dissipation coefficients R_ξ and R_η described in Section 6 are made to be zero everywhere at $l = 2$ through the scaling function S_l. However, near concave and convex corners, some dissipation has to be restored to maintain smoothness. This can be accomplished by introducing the blanking function $b_{j,k}$, which multiplies R_ξ and R_η at $l = 2$. The blanking function is zero everywhere except near concave ($\alpha_{j,k}$ or $\beta_{j,k} < \pi/3$) and convex ($\alpha_{j,k}$ or $\beta_{j,k} > 2\pi/3$) corners. For a concave or convex corner in the ξ-direction, we set

$$b_{j,k} = b_c, \quad b_{j\pm 1,k} = 0.5, \quad b_{j\pm 2,k} = 0.25, \tag{A.1}$$

while for a concave or convex corner in the η-direction, we set

$$b_{j,k} = b_c, \quad b_{j,k\pm 1} = 0.5, \quad b_{j,k\pm 2} = 0.25, \tag{A.2}$$

where $b_c = 1$ for a concave corner and $b_c = 0$ for a convex corner. At convex corners, dissipation at the neighboring points to the corner is still needed, but the dissipation at the corner point itself should be set to zero in order to produce the desired effects for the special schemes described in Section 8. In constructing $b_{j,k}$, it is assumed that successive corners in a coordinate direction are separated by at least four points. This is a reasonable assumption if one wishes to provide sufficient resolution for the flow around such corners.

REFERENCES

1. C. A. Atwood and W. R. Van Dalsem, Flowfield Simulation about the SOFIA Airborne Observatory. AIAA Paper 92-0656, 1992.

2. P. G. Buning, I. T. Chiu, S. Obayashi, Y. M. Rizk, and J. L. Steger, Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent. AIAA-88-4359-CP, *Proceedings of the AIAA Atmospheric and Flight Mechanics Conference*, Minneapolis, Minnesota, 1988.
3. J. Q. Cordova and T. J. Barth, Grid Generation for General 2-D Regions using Hyperbolic Equations, American Institute of Aeronautics and Astronautics Paper 88-0520, 1988.
4. S. K. Godonov and G. P. Prokopov, The use of moving meshes in gasdynamical computations, *U.S.S.R. Comput. Math. and Math. Phys.* 12:182–195 (1972).
5. D. W. Kinsey and T. J. Barth, Description of a Hyperbolic Grid Generation Procedure for Arbitrary Two-Dimensional Bodies. AFWAL TM 84-191-FIMM, Wright Aeronautics Laboratory, Wright-Patterson AFB, Ohio, 1984.
6. G. H. Klopfer, Solution adaptive meshes with a hyperbolic grid generator, in *Proceedings of the Second International Conference on Numerical Grid Generation in Computational Fluid Dynamics*, Miami, Florida, 1988, pp. 443–453.
7. W. Kordulla and M. Vinokur, Efficient computation of volume in flow predictions, *AIAA J.* 21: 917–918 (1983).
8. S. Nakamura, Marching grid generation using parabolic partial differential equations, in *Numerical Grid Generation*, (J. F. Thompson, Ed.), North-Holland, New York, 1982, pp. 775–783.
9. S. J. Parks, P. G. Buning, J. L. Steger and W. M. Chan, Collar Grids for Intersecting Geometric Components Within the Chimera Overlapped Grid Scheme, AIAA Paper 91-1587, in *Proceedings of the AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, Hawaii, 1991.
10. G. Starius, Constructing orthogonal curvilinear meshes by solving initial value problems, *Numer. Math.* 28:25–48 (1977).
11. J. L. Steger, Notes on Surface Grid Generation using Hyperbolic Partial Differential Equations, Internal Report TM CFD/UCD 89-101. Department of Mechanical, Aeronautical and Materials Engineering, University of California, Davis, 1989.
12. J. L. Steger and D. S. Chaussee, Generation of body-fitted coordinates using hyperbolic partial differential equations, *SIAM J. Sci. Statist. Comput.* 1:431–437 (1980).
13. J. L. Steger and Y. M. Rizk, Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations, NASA TM 86753, NASA-Ames Research Center, Moffett Field, California, 1985.
14. J. F. Thompson, ed., *Numerical Grid Generation*, North-Holland, New York, 1982.
15. J. F. Thompson, F. C. Thames, and C. M. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, *J. Comput. Phys.* 15:299–319 (1974).
16. A. M. Winslow, Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh, *J. Comput. Phys.* 2:149–172 (1967).

APPENDIX B

AIAA-93-0533

Development of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle

Daniel G. Pearce, Scott A. Stanley
Lockheed Engineering and Sciences Co.
Houston, Texas

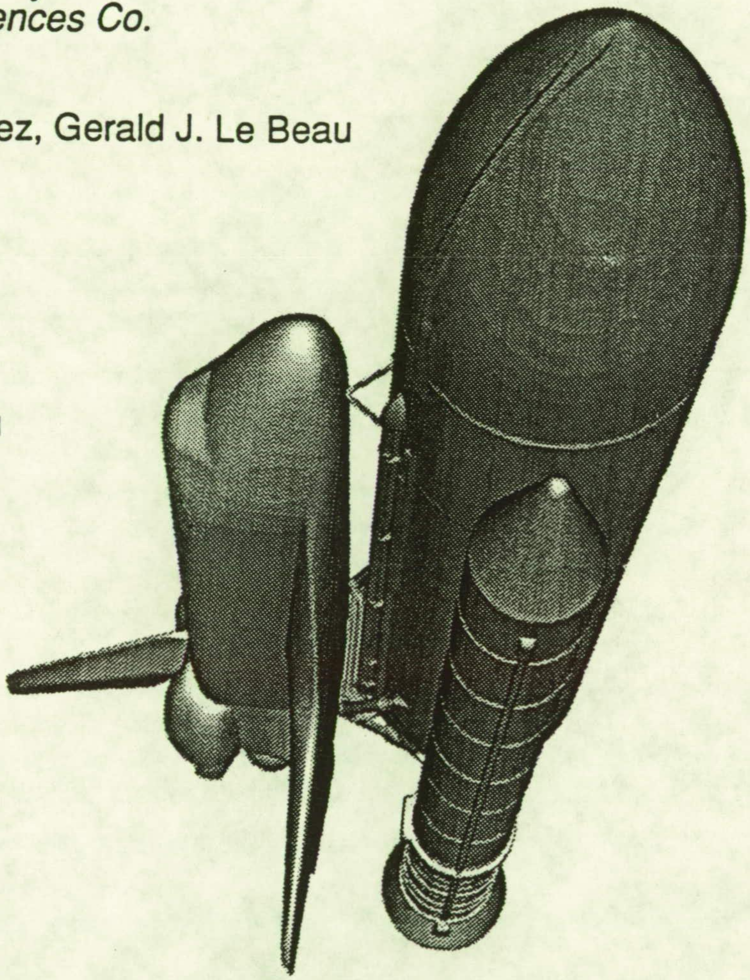
Fred W. Martin Jr., Ray J. Gomez, Gerald J. Le Beau
NASA Johnson Space Center
Houston, Texas

Pieter G. Buning
NASA-Ames Research Center
Moffet Field, California

William M. Chan, Ing-Tsau Chiu
MCAT Institute
Moffet Field, California

Armin Wulf
Control Data Systems Inc.
Berkeley, California

Vedat Akdag
Control Data Systems Inc.
Irvine, California



31st Aerospace Sciences Meeting & Exhibit

January 11-14, 1993 / Reno, NV

AIAA-93-0533

Development of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle

Daniel G. Pearce*, Scott A. Stanley**
Lockheed Engineering and Sciences Co.
Houston, Texas 77058

Fred W. Martin Jr.§, Ray J. Gomez§§, Gerald J. Le Beau§§§
NASA Johnson Space Center
Houston, Texas 77058

Pieter G. Buning+
NASA-Ames Research Center
Moffett Field, California 94035

William M. Chan†, Ing-Tsau Chiu††
MCAT Institute, NASA-Ames Research Center
Moffett Field, California 94035

Armin Wulf‡, Vedat Akdag‡‡
Control Data Systems Inc.
Berkeley, California 94704 *Irvine, California 92714*

ABSTRACT

The application of CFD techniques to large problems has dictated the need for large team efforts. This paper offers an opportunity to examine the motivations, goals, needs, problems, as well as the methods, tools, and constraints that defined NASA's development of a 111 grid/16 million point grid system model for the Space Shuttle Launch Vehicle. The Chimera approach used for domain decomposition encouraged separation of the complex geometry into several major components each of which was modeled by an autonomous team. ICEM-CFD, a CAD based grid generation package, simplified the geometry and grid topology definition by providing mature CAD tools and patch independent meshing. The resulting grid system has, on average, a four inch resolution along the surface.

1. INTRODUCTION

This paper emphasizes the culmination of many separate grid generation efforts that, together, provided a grid system for the Space Shuttle Launch Vehicle (SSLV) that has a very high degree of geometric fidelity. The distributed nature and large scale of this project provided special challenges that are likely to become increasingly typical of Computational Fluid Dynamics (CFD) analysis.

The SSLV has an extremely complex geometry for an aerodynamic vehicle. The four main SSLV components, the Orbiter, the External Tank (ET), and two Solid Rocket Boosters (SRBs), are sketched in Figure 1. The proximity of these bodies makes accurate modeling of the flow field and aerodynamic loads extremely dependent on the accuracy of the geometric model.

Initial efforts to model the flow field of the SSLV began with a coarse grid that severely simplified the geometry of the flight vehicle[1,2]. The importance of accurate geometry representation became readily apparent as geometry modifications improved comparisons between simulation results and wind tunnel data[3-5]. For example, one relatively small modification that added the integrated electronics assembly and attach ring to the SRB caused the simulated flow under the Orbiter wing to change enough to increase the wing-load by approximately twenty-seven percent of the maximum allowable load[3]. Several other components of the flight vehicle geometry, such as the ET/SRB and ET/Orbiter attach

---

[*] Senior Engineer, Member AIAA.
[**] Engineer associate sr., Currently pursuing a Ph.D. at U. C. San Diego, Member AIAA.
[§] Senior Engineer, Member AIAA.
[§§] Engineer, Member AIAA.
[§§§] Engineer.
[+] Research Scientist, Senior Member AIAA.
[†] Research Scientist, Member AIAA.
[††] Research Scientist, Member AIAA.
[‡] Consultant.
[‡‡] Consultant. Previously of Rockwell at Downey.

hardware, also block the flow between the Orbiter, ET, and SRBs. Unfortunately, the crudeness of previous grid generation tools and lack of concise and accurate geometric definitions hampered the refinement of the "preliminary" grid system. Much of the preliminary geometric model had been defined with available drawings and photographs that were often lacking in detail. These approximate surface definitions had been encoded and discretized within grid-specific FORTRAN programs. Typically, the resulting grids were then exposed to a tedium of small perturbations and "smoothing" operations as a final grid was evolved. This "coding and evolution" process was too time consuming, cumbersome and laborious to be considered for major refinement of the preliminary model.

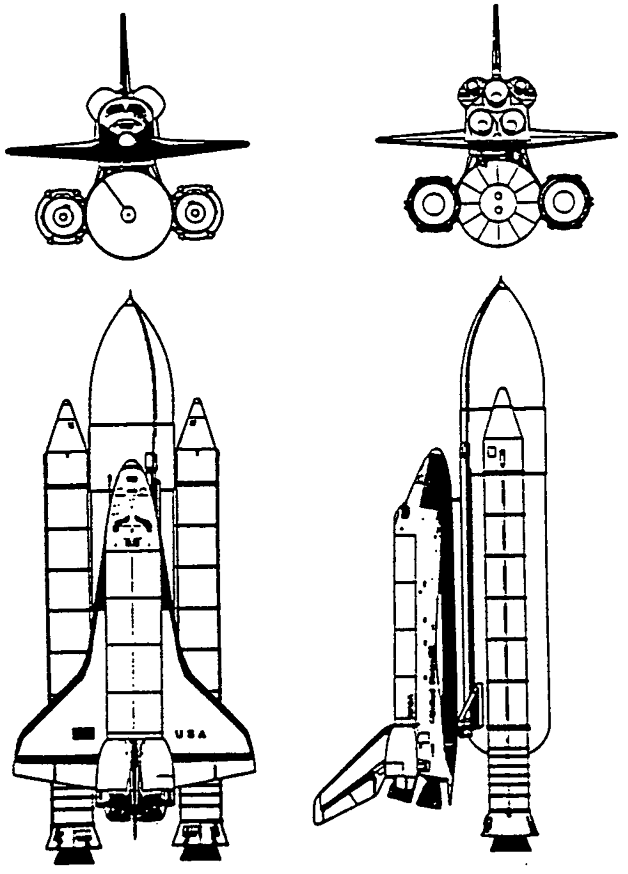


Figure 1. Front, Rear, Top, and Side view drawings of the SSLV configuration.

The development of the present enhanced-resolution SSLV grid system began with a study of geometric modeling and grid generation requirements. Evaluation of available tools and data showed the desirability of a grid generation tool that could utilize the mature capability of Computer Aided Design (CAD) techniques and geometric databases. Several subcomponents of the launch vehicle had already been modeled, with many man-years of effort, by other groups and were available. Furthermore, these CAD models represent the most accurate and complete geometric descriptions available for the launch vehicle. Several different CAD-to-grid generation tools were evaluated but most had only limited abilities for generating ordered meshes. Control Data Systems' software package, ICEM-CFD[6,7], provided the only CAD interface for grid generation that allowed an ordered mesh to be generated independent of the orientation of the underlying geometric patches.

The grid system was generated for use with the flow solver OVERFLOW[8] that utilizes the Chimera domain decomposition approach[9-12]. The grids for each geometric component of the multi-body geometry were generated independently and overlapped to produce a complete grid system. During flow simulation the boundary values of each overlapping grid are updated through interpolation. Only the local resolution and amount of overlap between grids need be considered when interfacing two adjacent grids. Grid generation is not limited by the point-for-point or cell-face matching associated with "Block Zone" methods. This allowed a high degree of flexibility in determining the resolution and topology appropriate for modeling each component of the geometry.

As with many other large-scale CFD simulation efforts, the Space Shuttle Launch Vehicle ascent simulation was accomplished in a team effort that was spread over several locations and organizations. Although full group discussions were conducted periodically, it was virtually impossible for each member to remain cognizant of all decisions. It proved vital to have a method of breaking the problem into separate tasks with limited interfacing requirements so that each group was encouraged to progress autonomously whenever possible.

The approach, criteria, tools, and limitations associated with the enhanced-resolution SSLV grid system are discussed below in the geometric definition and grid generation sections.

2. GEOMETRY DEFINITION

The search for data that defined the outer mold line surface geometry of the Space Shuttle Launch Vehicle revealed several sources including: Space Shuttle Master Dimension Specification[13], Interface Control Document # ICD-2-00001[14], numerous photographs, literally thousands of drawings and several CAD and numerical descriptions. Responsibility for subcomponents and related data was scattered over several NASA centers and contractors. Although this information was available for use by NASA, creating a concise and appropriately accurate model posed to be a major undertaking. The level of detail varied from source to source. Manufacturer's production drawings represented the most accurate definition of the actual vehicle components but were far too numerous to compile for an entire vehicle in the time allotted. The Master Dimension Specification contains a detailed description of the Orbiter but left the ET, SRB, and protuberance definitions to be described by ICD-2-00001. Both documents were encumbered with a complicated revision process where a current surface geometry could only be determined by superimposing all previous revisions. However, these documents did provide accuracy appropriate for spot checking and verifying other models. The decision to use a CAD based grid generation package provided a medium for compiling definitions and a means of utilizing existing databases that were previously inaccessible. CAD information was collected from several institutions and compiled into major component CAD models representing the Orbiter, External Tank, and Solid Rocket Boosters. These three major component models were verified, and supplemented where necessary with information from drawings, photographs, and flight vehicle inspections.

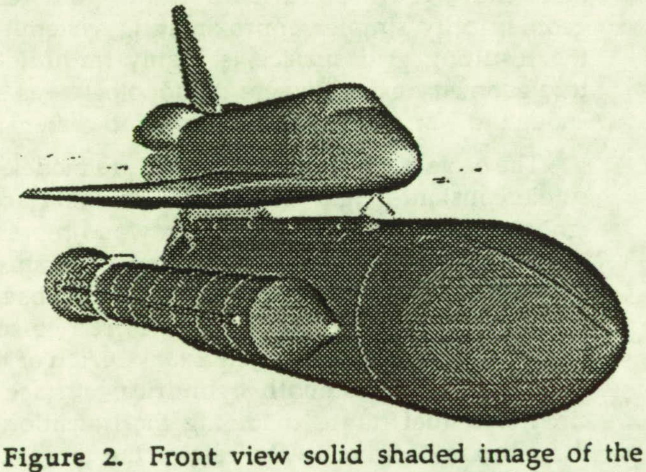


Figure 2. Front view solid shaded image of the enhanced resolution SSLV surface grid. Note the resolution of the cable trays and feed line on the ET, the circumferential plant and field joints and cable tray on the SRB, and the Orbiter canopy and forward bipod.

Each of these major components was developed independently with only occasional updates between groups. Geometry definitions of the separate protuberances and components of the ET were created by Martin Marietta at Michoud. The highly detailed CAD descriptions of the ET were modified to remove internal features, converted to surface models, and composed into an outer moldline surface model by Lockheed Engineering and Sciences Company at Houston, Texas. CAD definitions of portions of the Space Shuttle Orbiter geometry were created by Rockwell International Space Systems Division at Downey, California and at Palmdale California. Composition of the Orbiter models into a CAD model suitable for CFD was performed by Rockwell-Downey and NASA Ames Research Center. The SRB geometry definition was created and composed with cooperation amongst Rockwell-Downey, NASA Ames, and NASA Johnson Spacecraft Center. The following is a detailed discussion of the efforts associated with each major component.

2.1 External Tank

The true surface geometry of the ET includes many protuberances such as feedlines, cable trays, fairings, and large structural members (for attaching the ET to the Orbiter and SRBs). Each of these protuberances is exposed to the exterior flow field during launch, and many present large cross sections normal to the flow field. The enhanced-resolution flow field simulation needed to be accurate enough to compare well with flight load and flow field measurements. A target surface resolution of four inches for the overall launch vehicle model was determined largely by the resolution required to capture the effects of these ET protuberances.

Defining the geometry of all tank protuberances greater than four inches using the tools and data available for the preliminary grid system would have been an insurmountable task. The complexity of the geometry, the continuous revision process, and the accuracy required made it difficult to identify and locate accurate and current information. This was exacerbated by the difficulty in defining the nonuniform layer of protective thermal foam over each component of the ET. Some outer moldline definitions were unavailable due to the operational approach of defining the thickness of the foam with large tolerances and allowing the line worker to define the final shape.

Fortunately, Martin Marietta at Michoud had already undertaken an effort to update the definition of each protuberance of the ET. The resulting models were extremely detailed three-dimensional wireframe CAD definitions. Each outer moldline definition included descriptions of the underlying metal structure and each layer of thermal protection foam that is applied during assembly of the ET.

A formal request was made for the electronic transfer of these protuberance models and a description of the tank surface itself from Martin Marietta to NASA JSC. The transfer required a conversion from the Computer Vision CAD system, at Martin Marietta, to the Control Data System ICEM-CFD CAD system, at NASA-JSC. This was accomplished using IGES format translators within each code (see section 2.4). To expedite the transfer, "clean up" of the models (to remove internal details and small elements such as bolts and lockwire that were obviously below the anticipated resolution of the computational grid) was limited to those features that were easily removed prior to transfer.

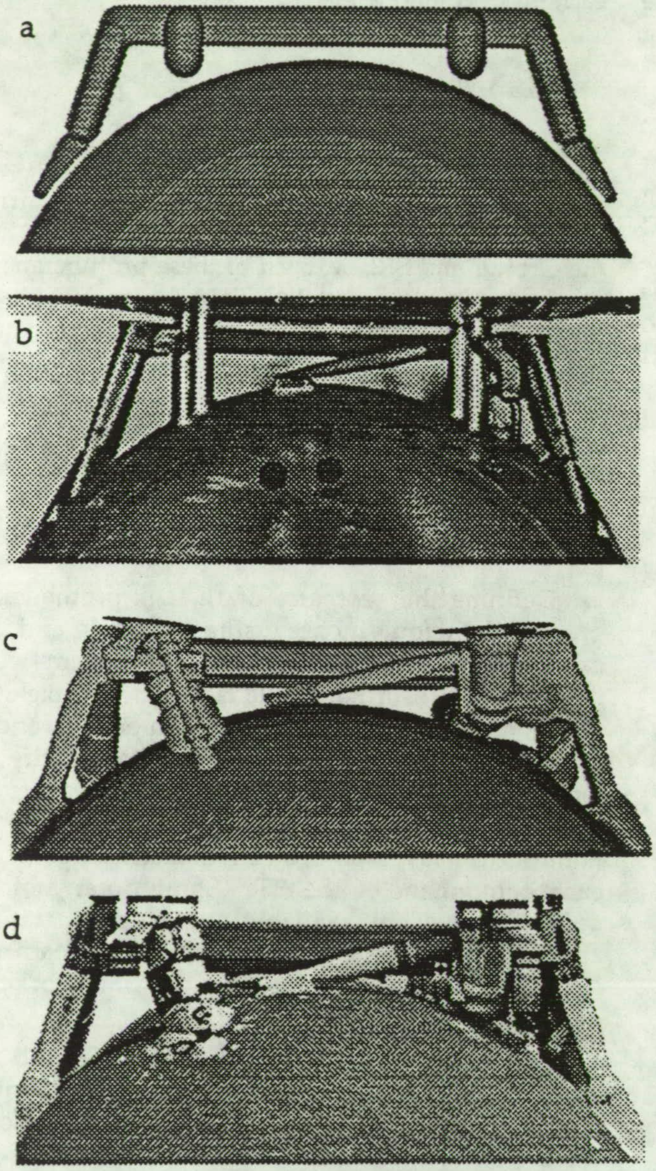


Figure 3. A rear view comparison of three models and the SSLV ET/Orbiter aft attach structure. The subfigures are: a) a solid shaded image of the preliminary grid system, b) a photograph of the IA-613 wind tunnel model, c) a solid shaded image of the enhanced resolution grid system, and d) a photograph of the actual flight hardware.

As the tank subcomponent models were received, the wireframe definitions were used to create an outer moldline surface model of the ET and the ET/Orbiter attach hardware. In the early stage of the redefinition, the surface geometry was not modified beyond the level convenient for creating the component surfaces expediently. The outer mold line surface geometry was generally left as it was originally defined in the Martin Marietta CAD definition. Considerable effort was required to get all components into a common coordinate system and to cycle through the inevitable inconsistencies encountered when assembling any large collection of parts.

The intermediate result of collecting these models was a very detailed CAD surface model. A considerable amount of surface approximation was required to remove fine surface details and simplify the topology to allow discretization with a reasonable number of points while maintaining the targeted four inch surface resolution. However, the process of eliminating unwanted detail resulted in a different approximation than would have been achieved if an approximating surface had been created from scratch. It was easier to retain detail and the accurate surfaces than to create a geometrically simpler approximating system. Thus the resulting grid surface is highly faithful to the true geometry except where the topology was severe enough to warrant the effort required to change it.

The cable trays on the ET surface are modeled as a single constant-height ridge in the tank surface that ignores the many separate cable tray and repress line tie down brackets. The LO2 cable tray (that runs from the intertank region, below the Orbiter nose, to a fairing at the forward tip of the ET) can be seen in Figure 2. The multi-faceted intertank region of the ET is modeled as a smooth cylindrical surface with circumferential ramps modeling the transition from the intertank region to the rest of the tank surface. The LO2 feed line, tie down brackets, and forward fairing is modeled as a bump in the ET surface that can be seen in Figure 2 beginning below the Orbiter nose and running back to the aft attach hardware. Figure 3 shows a progression of fluid dynamic models of the ET/Orbiter aft attach hardware model in comparison with the flight hardware. The preliminary (symmetric) grid system, shown in Figure 3a, captures most of the large elements that block the flow but does not include the intersections between bodies. The wind tunnel model, Figure 3b, includes the diagonal strut and umbilical cable trays but underestimates the size of each component by the thickness of the thermal protection foam. The enhanced resolution grid system, Figure 3c, can be seen to be the most accurate model of the actual flight aft attach hardware, Figure 3d. Comparing the LH2 feed line region model on the left side of Figure 3c to the flight hardware in Figure 3d shows that virtually all the features are represented. However, the LO2 region, on the right side of Figure 3c, is missing some detail. The difference between these two models reflects that on the LH2 side each component was relatively exposed while many of the features on the LO2 side were so tightly clustered that many were sheltered from the oncoming flow. As a result, some components on the LO2 side were deleted from the model and several were combined to form one LO2 "blob" by merely filling in the small remaining gaps. In effect, the approximation fits the actual surface like shrink-wrap.

2.2 Orbiter

The Space Shuttle was designed in the early seventies using parametric equations to define the cross sections of the Orbiter outer mold line at discrete locations along the fuselage. The Numerical Design group at Rockwell-Downey maintains this geometric database via the Orbiter Master Dimension Specification Program. The output of this program yields point data that models a smooth surface description of the Orbiter that is devoid of tile gaps and depressed regions where tiles have been replaced by thinner thermal blankets. Also missing are the variations found in each of the Orbiter vehicles due to manufacturing tolerances. Both the preliminary and enhanced grid surface models of the Orbiter were initially based on surface points supplied by Rockwell's Numerical Design Group.

The preliminary grid system was developed from surface points that defined the Orbiter cross section at 107 specific stations, or approximately every ten inches, in the streamwise direction. A FORTRAN program was written to create spline curves at each streamwise station and to interpolate between the splines to create surface grid points in a structured mesh. This model was accurate but lacked detail towards the rear of the Orbiter and the surface grid topology was difficult to manipulate.

The initial enhanced-resolution CAD geometry was developed at Rockwell-Downey, using ICEM-CFD, from a much larger set of surface points. Point data was obtained at each half inch interval along most of the fuselage and at each one hundredth inch interval near regions where the geometry changed rapidly such as the nose, canopy, Orbiter Maneuvering System (OMS) pods, and elevon regions. B-splines were created from select point data at each cross section and B-spline surfaces were created from select curves. Sixteenth order surfaces and tight tolerances were used to insure accuracy. Sharp corners were modeled with multiple segment B-splines. This tedious interactive process has now been automated within ICEM-CFD. The resulting model was smooth and fairly free of surface undulations.

However, the need for accurate definitions of the elevon gaps and wing tips led to a search for even more accurate models of these regions. An effort was underway to create a CAD description of portions of the Orbiter surface at NASA Kennedy Spacecraft Center and Rockwell-Palmdale to support tile fabrication activities. A formal request was made to obtain the CATIA CAD system models in IGES format. These detailed models were used to improve the CAD definition of the Orbiter lower wing surfaces, elevons, elevon flipper doors, elevon gaps, and wing tips.

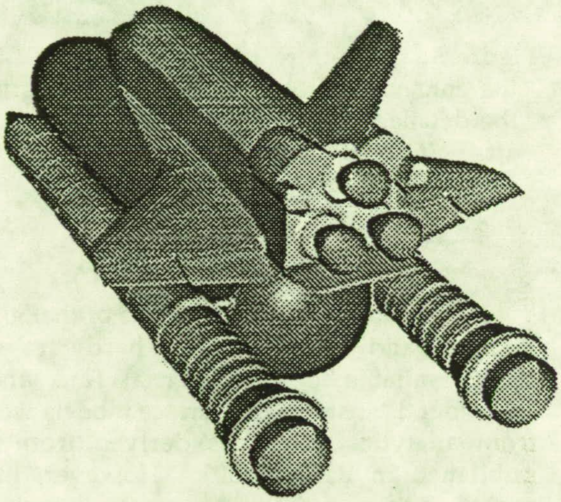


Figure 4. Rear view solid shaded image of the enhanced resolution SSLV surface grid. Note the resolution of the elevons and flipper doors on the Orbiter wings and the many concentric plant and field joints near the base of the SRBs.

Orbiter base components such as the Space Shuttle Main Engines (SSMEs) and the body flap were created by various members of the team and combined into a consistent model at NASA Ames and Rockwell-Downey. Cooperation between these sites was greatly enhanced by the use of a common CAD package. The resulting enhanced grid system surface for the Orbiter is presented in Figures 2,4, and 5. The smooth forward portion of the Orbiter and canopy can be seen in Figure 2. Figure 4 shows the complexity of the rear of the Orbiter wings and base. The wing models include inboard and outboard elevons, gaps, and upper surface flipper doors. Figure 5 provides a close up view of the Orbiter base. Note that the SSMEs, OMS pods, and Reaction Control System (RCS) extensions are all included. The OMS engine bells were excluded since they do not fire during ascent and are located in the separated region caused by the OMS pods.

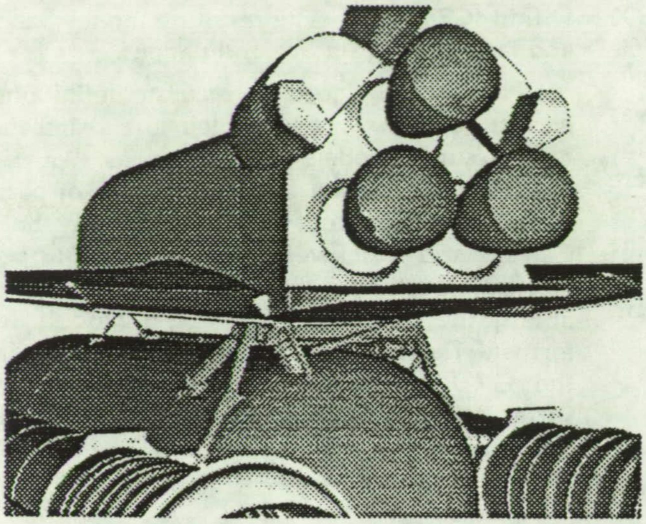


Figure 5. Close-in rear view solid shaded image of the enhanced resolution SSLV surface grid. Note the detailed model of the Orbiter base region and aft ET/Orbiter attach region.

2.3 Solid Rocket Booster

No detailed CAD descriptions of the Solid Rocket Boosters and ET/SRB attach hardware similar to those available for the External Tank and Orbiter were found. Instead, the surface models were created from analytic definitions derived from drawings published in ICD-2-00001. However, most large protuberances on the SRB surface are included: The segmentation rings (plant and field joints) can be seen in each of Figures 2,4 and 5; The external cable tray can be seen in Figure 2; while the integrated electronics assembly box can be seen best in Figure 5. The current grid system models the Redesigned Solid Rocket Booster (RSRB) and work is in progress to model the Advanced Solid Rocket Booster (ASRB).

2.4 IGES Interface

The Initial Graphics Exchange Specification (IGES) provided a standard interface between CAD systems, that was reliable *most* of the time, from several different manufacturers. The transfers of wire frame CAD models encountered very few problems as points, lines, arcs, curves, etc., were translated to and from IGES format. However, increasing numbers of surfaces, which have not been established within IGES, were included in later models. Trimmed surfaces and surfaces of revolution were particularly difficult to transfer consistently. Extremely high order surfaces also caused problems when the capability of the receiving system was exceeded. In most cases, the offending surfaces were merely deleted by the ICEM-CFD translator and required recreation. Close contact with the vendor and Control Data Systems personnel provided work-arounds and custom fixes for specific problems.

3. GRID GENERATION

The capabilities of ICEM-CFD and Chimera domain decomposition combined to greatly improve the standard approach to grid generation. In this project, ICEM-CFD was used primarily to develop surface geometry and surface grids. User interaction with the CAD interface of ICEM-CFD greatly reduced the cycle time in the iterative process of defining and refining entities that control the geometry and topology of the grid. Volume grids were generated with the hyperbolic grid generation program HYPGEN. The three-dimensional flow field domain was decomposed into overlapping regions in a Chimera approach that allowed each geometric component of the launch vehicle to be modeled in a separate effort. The program PEGSUS was used to interface the resulting overlapped volume grids. HYPGEN and PEGSUS are described in Sections 3.2 and 3.3, respectively.

3.1 Surface Grid Development

One key advantage of ICEM-CFD is its ability to develop a grid without being constrained by the orientation or extent of the underlying network of geometric patches. This "patch independent meshing" decouples the process of defining the geometry from the process of defining the orientation and distribution of the surface mesh. The creation of the CAD definition can proceed in a manner convenient for modeling geometric features while the topology and density of the mesh can be overlaid according to the needs of flow field features.

Once the CAD surface geometry is complete a computational mesh can be defined. The interactive grid preprocessor module, MULCAD[7], within ICEM-CFD uses a combination of base level CAD tools and menu prompts within a subprocess of the CAD environment to define the extent, topology, and clustering of the grid. The user has the convenience of alternating between geometric manipulation and the set up of grid control elements. This can be particularly useful in simplifying detailed geometric models to a level appropriate for the given discrete resolution and budget of grid points.

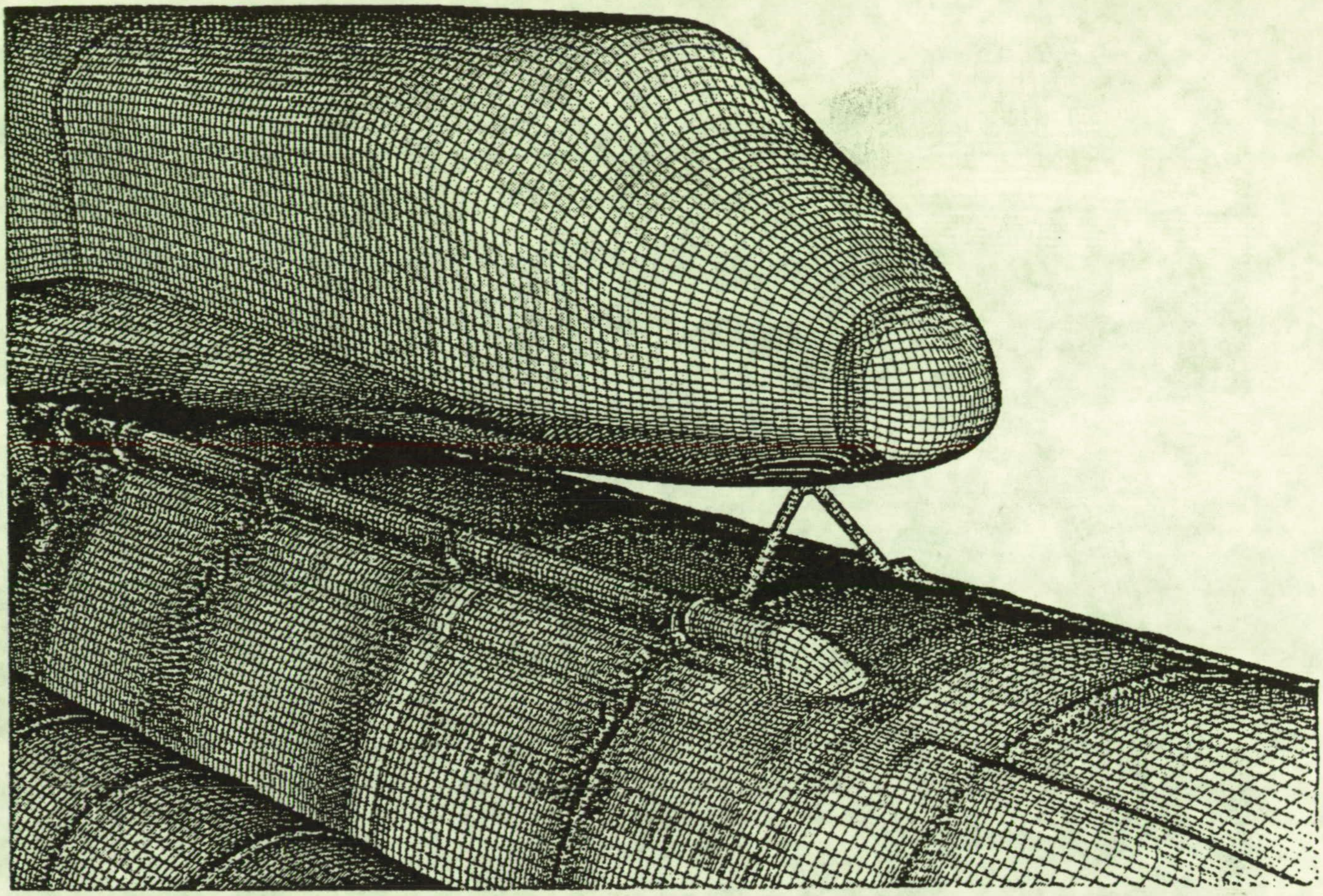


Figure 6. Close up view of the surface grid of the SSLV showing the transition from an H-type to an O-type grid topology near the Orbiter nose. Also note the details captured within the ET surface grid.

In this phase of generating an ordered mesh with MULCAD, a second network of geometric elements is created over the original geometry. The layout of this second network can be defined with the same CAD tools that were available for the geometry but oriented to explicitly control the grid topology. This approach complimented the capabilities of the Chimera domain decomposition scheme by allowing varied topologies to be overlaid on the same geometric model. Figure 6 shows an H-grid topology on the nose of the Orbiter smoothly transitioning to the O-type topology on the Orbiter body.

The distribution of points along each control element is defined interactively and can be viewed for qualitative verification within MULCAD prior to computing the grid. The resulting hierarchy of grid control elements: grid control curves (called Edges) combine to define grid control surfaces (called Subfaces) which combine to form volume or surface block definitions (called Domains). Typically, curves available from the preceding geometry definition phase are supplemented with curves created specifically for grid control then combined and given control attributes within MULCAD. The usual varieties of tools are available for distributing points along curves and surfaces, and within volumes.

Experience with the preliminary grid system dictated which geometric components needed to be included in the enhanced-resolution grid system. The resolution required to include these components defined the four inch nominal grid resolution for the grid system. Of course, the grid was clustered tighter than nominal near regions with complex geometry and/or flow field features and relaxed near smooth geometry or low gradient regions. The final grid system has a surface resolution that typically varies from two to twelve inches. A minimum of three points were used to resolve geometric slope discontinuities, and five points were used where possible. Shocks are resolved by the flow solver with five grid points and central differencing with 2nd- and 4th-order smoothing.

The large number of surface grid points required to capture the SSLV geometry is evident in the representation of the ET in Figures 6,7. In some cases, the number of points required to capture a geometric

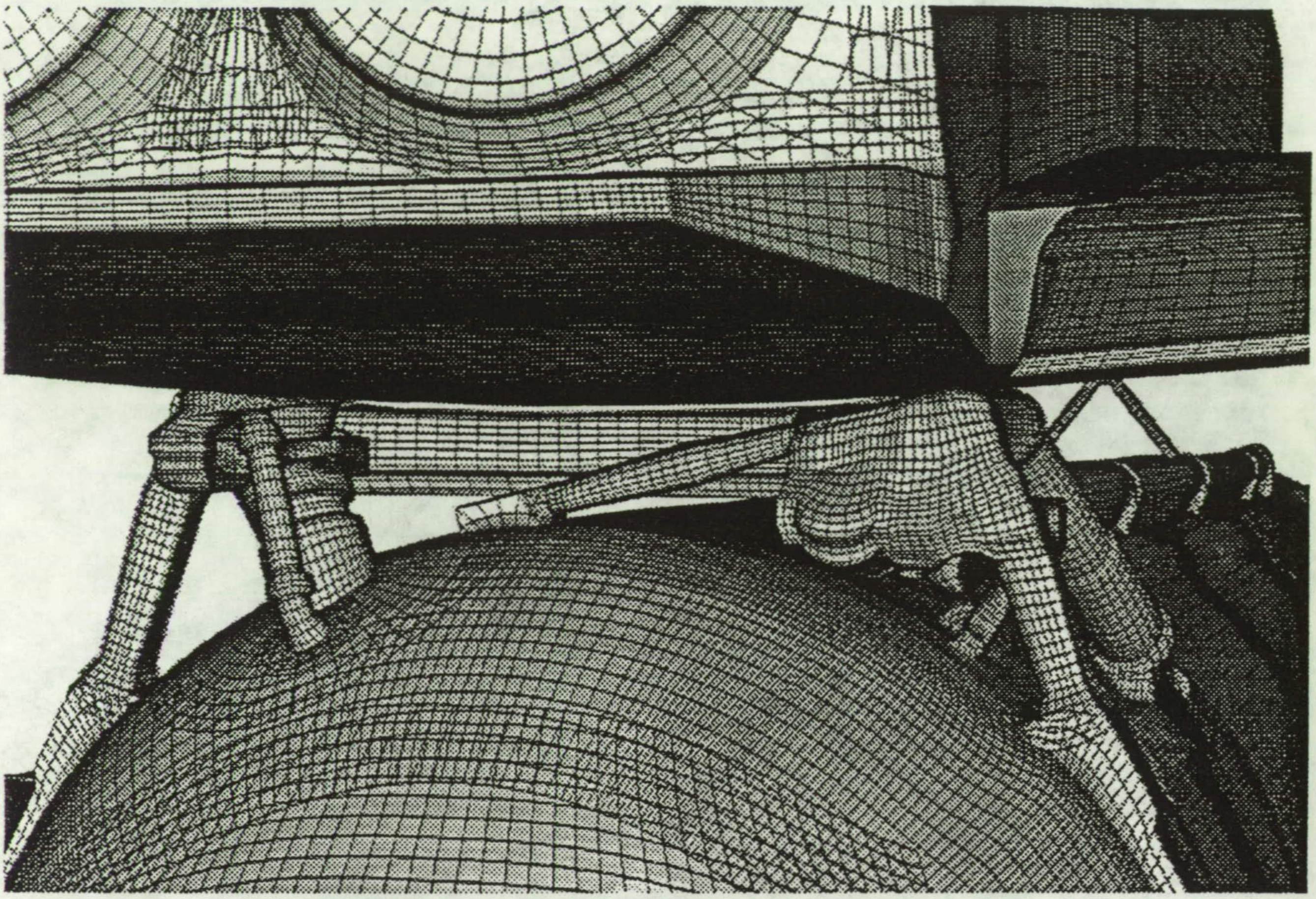


Figure 7. SSLV surface grids in ET base region. Note how the domain decomposition simplifies the topology of each individual element and restricts the areas affected by local increases in resolution.

element exceeded the number warranted for its inclusion. As a result, the geometry and/or topology was often simplified and the preceding processes were repeated. Components immediately downstream of larger components were assumed to be in separated flow regions that would cause them to have small effects on the flow field. Therefore the grid system most accurately depicts those that are directly exposed to oncoming flow. In general, any component smaller than a four inch diameter was not considered resolvable.

Upon completion of the definition phase of surface mesh development, an input file is written from MULCAD. PADAMM[7], the grid generation module for the ICEM-CFD package, is a batch program that uses the input parameters and grid control element geometry from MULCAD to generate an intermediate mesh and "project" it to the underlying CAD surfaces in a proprietary process. PADAMM provides both algebraic and elliptic tools for generation and smoothing of the projected surface grids.Experience showed that the resulting mesh faithfully followed the geometry of the CAD surface model but that special control was needed to maintain the intended distribution of points in regions of high curvature. When the control geometry oversimplified the underlying CAD surface model the resulting grid was not smoothly distributed. Additional control lines were added to simplify the projection by breaking the geometry into simpler pieces. However, neither the control line geometry nor the point distribution along the control line is adjusted by PADAMM during grid generation. Unfortunately, this effectively requires the user to guess the exact location of the controlling grid line within the mesh which can be a time consuming iterative process.

Originally, PADAMM was executed as a batch process separate from the CAD environment. Towards the end of this project the ability was added to execute PADAMM from within ICEM-CFD. This reduced the cycle time for interacting with the geometry, topology, and distribution of the grid. The capability to generate volume grids within ICEM-CFD was not utilized in favor of the hyperbolic grid generator described in the section below.

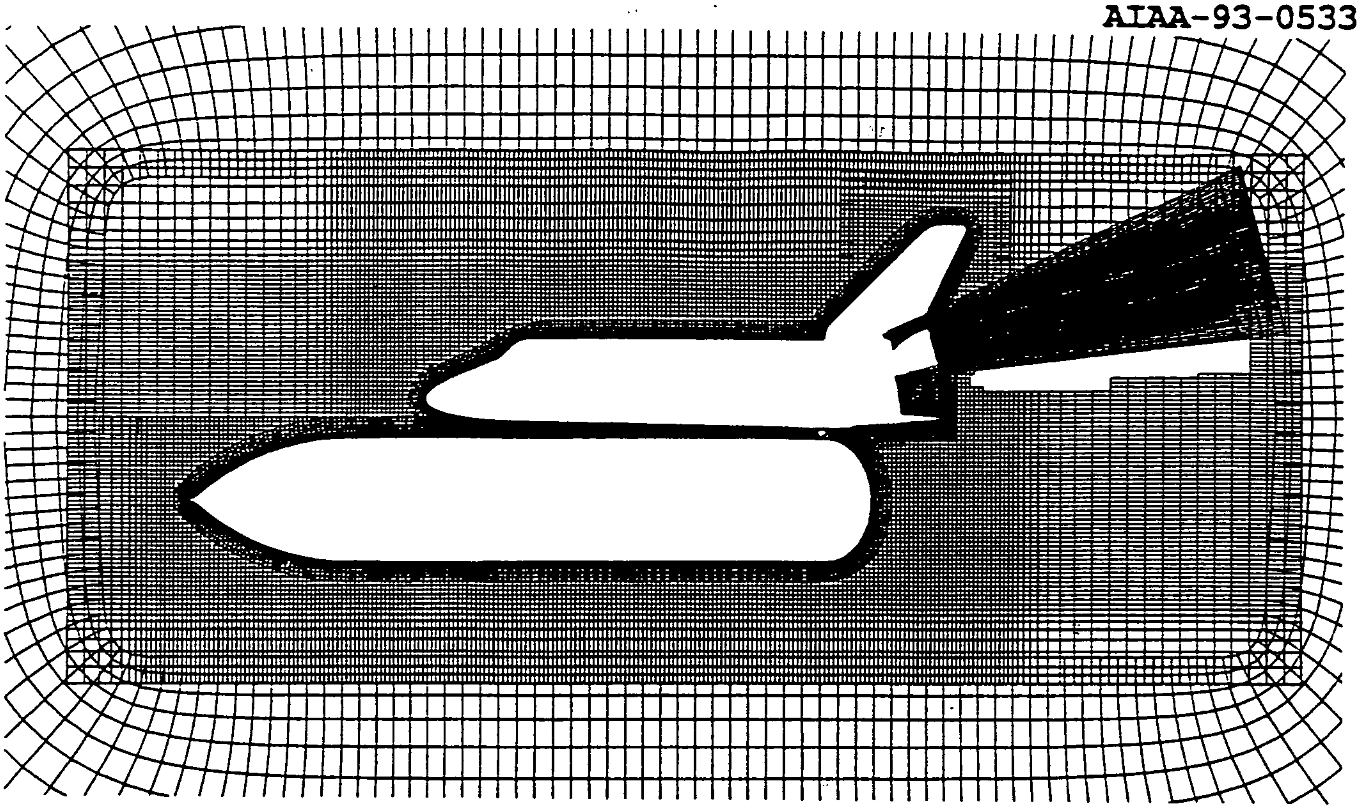


Figure 8. Side view of the SSLV grid system with volume grids plotted on the pitch plane. Note that the surface oriented grids extend only a short distance before the medium resolution rectangular box grids form a transition to the coarse far field grid. The blank region behind the Orbiter is discretized by the SSME plume grids that do not lie in the pitch plane and were not plotted.

3.2 Volume Grid Development

The hyperbolic grid generator HYPGEN[15] developed at NASA Ames Research Center was used to create the volume grids for several reasons. The hyperbolic marching scheme required one to two orders of magnitude less CPU time than typical elliptic methods. The user effort to set up the volume grid generation process was focussed on producing a high quality surface grid rather than being divided to define all six faces of the volume control block as with elliptic and algebraic schemes. The release of control over the extent and shape of the outer boundaries required by the hyperbolic method was appropriate for generating grids for the Chimera grid system. No constraints are imposed on the shape or location of grid outer boundaries beyond maintaining a one-cell minimum overlap between neighboring grids. On the other hand, tight control of clustering and orthogonality, which is provided by hyperbolic methods, was required for the grids to be used with the thin-layer Navier-Stokes approximation in the flow solver OVERFLOW. The enhanced robustness of HYPGEN allowed high quality grids to be produced with very few attempts for even highly irregular and nonconvex geometry.

The user interface, UI[16], provides a convenient display of HYPGEN control parameters on a Silicon Graphics workstation and speeds the development cycle time. Default values for the input parameters are automatically initialized and can be easily modified. UI can be run entirely locally on the workstation or the remote processing option can be invoked to allow the CPU intensive grid generation process to be run on a faster remote machine such as a CRAY. (This reduced the execution time for generating a volume grid from nearly an hour to a few seconds.) Viewing the resulting grid is provided by UI by creating a window running PLOT3D[17] and command files for setting the appropriate viewpoint and displaying default surfaces.

Typical inputs for SSLV grids with no-slip surfaces were: a first cell normal spacing of .00016 inches, an exponential distribution in the normal direction with a constant stretching rate in the range 1.25 to 1.29, and a maximum normal extent of 20 to 60 inches. The very small normal spacing corresponds to $y^+=1.0$ at ten percent aft on the launch vehicle with a flight Reynolds number of 2.7×10^8. A higher stretching rate and normal extent was used with very smooth geometry and somewhat lower values were required for stable operation of HYPGEN with the more irregular surface elements.

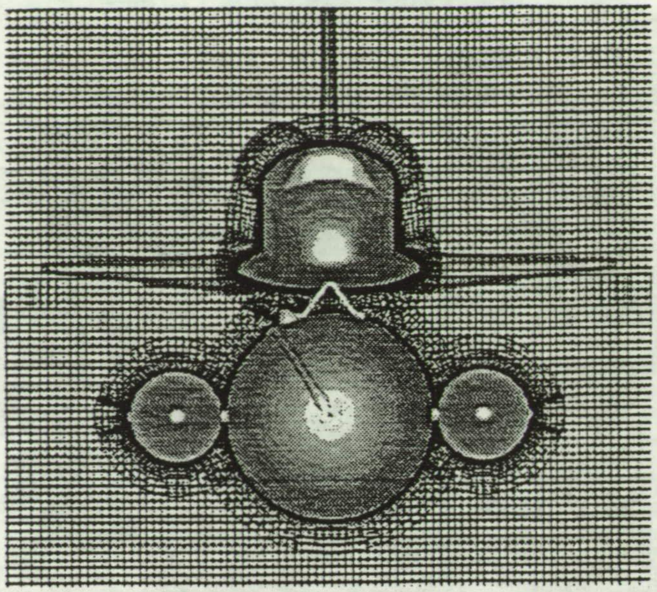


Figure 9. Front view of the SSLV grid system with volume grids plotted on a plane midway along the fuselage. Note again the short extent of the surface fitted grids.

Figures 8 and 9 illustrate that the hyperbolically generated surface oriented grids can have topology and outer boundary shape independent from the surrounding domains. This made volume grid generation one of the quickest tasks.

In some cases, a surface grid was incompatible with HYPGEN and a volume grid could not be generated properly. Typically, this could be easily resolved by simplifying the underlying surface or surface grid

3.3 Domain Decomposition and Interfacing

In order for a flow solver to update a solution on the interior of a grid each point on that grid's boundary must be provided with a boundary condition. In the case of those points that overlap other grids within the Chimera grid system interpolation is used to update the appropriate flow values. The grid interfacing program PEGSUS[18], developed by CALSPAN at Arnold Engineering Development Center, preprocesses the grid system to determine the appropriate information to be provided to the flow solver for updating each grid boundary point of the system during simulation. An interpolation cell that surrounds the boundary point is located in a neighboring grid and bilinear interpolation constants are calculated based on the relative location of the boundary and interpolating cell points. The resulting "stencil" information is stored for use during flow simulation. PEGSUS was modified at NASA Ames to provide for the demands of the viscous spacing and to accelerate processing time.

The first step in dividing the flow field was to determine how to divide the complex surface geometry into topologically simple regions. Each region was made as large as possible to minimize the number of separate grids and interface overhead required to model the geometry. Larger grids also increase vector length and vector speed-up during flow solution. On the other hand, the 64 megaword in-core memory limit, and the 16mw optimum operational queue size of the Cray-Xmp computer at NASA JSC limited the number of points that should be included in each grid.

Early estimates of the requirements of the flow solver OVERFLOW indicated that thirty-two words of memory were required per grid point. Thus the maximum volume grid size was limited to 500,000 points. The tight initial cell spacing, cell stretching criteria, and normal extent target limited the maximum number of surface points that could be included in each grid to approximately 15,000. This number and the nominal four inch spacing criteria determined the surface area that could be modeled with one surface grid domain.

The actual decomposition of a surface model into overlapping surface grids was further adjusted to reflect local detail and grid interfacing criteria. In the aft attach region grid sizes were limited to simplify the topology of each grid, see Figure 7. The struts, feedlines and cross members of the aft attach region were modeled as O-type grids that wrapped around the geometric element and flared out onto adjacent elements to provide the required overlap. This allowed grids to be produced with viscous clustering in only one index direction as is appropriate for the thin-layer Navier-Stokes flow solver.

When two adjacent grids have similar resolution and topology where they overlap the Chimera scheme is merely a generalization of the Block Zone approach. This simple overlap can be easily constructed for simple geometry and provides a quick method of decomposing large smooth areas into appropriately sized grids with reliable interfaces. Higher local resolution can be utilized in one grid while maintaining similar topology at the interface. However, a change in resolution across interfaces is best limited to a factor of two especially near high gradient regions to avoid excessive interpolation errors in determining boundary values during the flow solution.

It can be cumbersome to maintain simple overlaps between grids where the topology of the geometry is complicated. One key advantage of the Chimera grid scheme is that the topology of each grid is really independent. Topologically simple grids can be used without regard to the surrounding system as long as the resolution at the interface between grids is comparable. This is exemplified by the rectangular box grids, that form the interface between the complicated surface oriented grids and the simple free field grid, see Figures 8-10. If grid points fall inside bodies or into regions where the resolution is not similar (and would not yield proper interpolation stencils) then the unwanted points are "blanked", flagged as not contributing, to the solution. The nearby points that are adjacent to the blanked region of the grid are treated as boundary points and updated through interpolation. The process of identifying points that should be blanked, "cutting holes", and the process of providing interpolation stencils can be accomplished using the program PEGSUS.

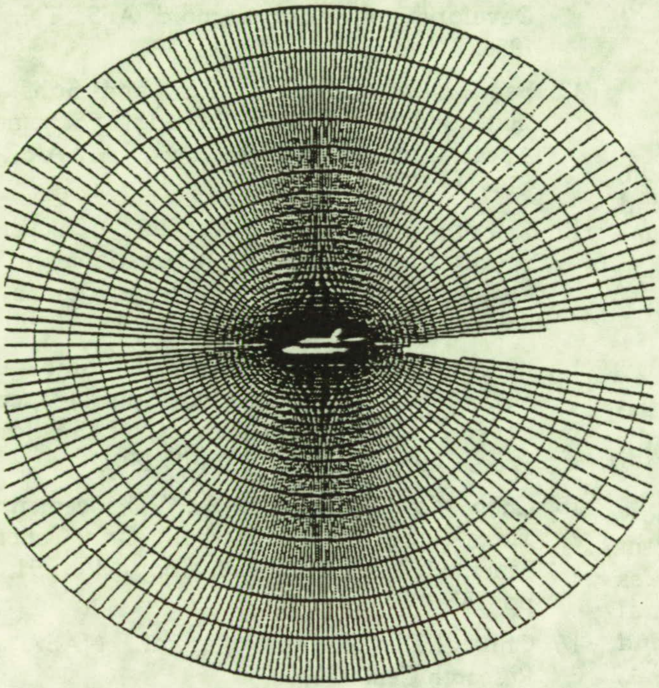


Figure 10. Side view of the outer domain of the enhanced-resolution SSLV grid system. The blank region that extends to the exit of the far field grid is discretized by the SRB plume grids that do not lie in the pitch plane and were not plotted.

The tools available within PEGSUS for cutting holes in a grid are based primarily on families of grid points from adjacent grids. Occasionally, "phantom" grids are created to cut holes that are not easily defined. These phantom grids are excluded from the resulting grid system since they do not contribute to the flow simulation.

As a rule, each grid was generated such that its outer and hole boundary points overlap neighboring grids by at least two grid cells. This insures that interpolation stencils would be based solely on points that were updated within the flow solver. Two overlapping stencils will continue to trade information that is independent of the surrounding flow simulation. A point that can be updated without relying on interpolated points is said to have a stencil with Quality, Q, equal to one. Quality values in the range $1>Q\geq0$ are assigned based on the relative location of the point to the interpolated points within the stencil. The number of times that PEGSUS searches for stencils and the minimum quality that will be accepted during each search can be defined in the input. The 16 million point enhanced-resolution grid has 1.2 million points that require stencils. Most stencils are of quality equal to one while 2.2% and 0.1% have qualities in the ranges $1>Q\geq0.6$ and $0.6>Q\geq0.2$, respectively. Stencils with Quality less than .2 would exchange little or no new information at the overlapping boundary during iteration. Points that failed the minimum quality test are labeled as "orphans". The process of manipulating the PEGSUS inputs to satisfy recalcitrant orphans can be time consuming.

In practice, it can be difficult to ensure good overlap with three-dimensional grids that have varying spacing and topology, especially near holes. If acceptable interpolation stencils could not be established, it was necessary to loop back to previous steps to modify the resolution, extent, and orientation of the grids. Test cases to gain experience in developing models that provided compatible interfaces helped to minimize the time spent reproducing the underlying grids.

A serious impediment to the timely completion of this grid system was the lack of an interactive environment for the interfacing procedure. Roughly half of the grid system development time was consumed in this phase of the project. The relation between the three-dimensional hole boundary points and the surrounding grids was not always easily visualized. PLOT3D running on an IRIS graphics workstation was severely taxed even with its color display capabilities and fast local rotation. A three-dimensional viewing device would have been very helpful.

CONCLUSIONS AND RECOMMENDATIONS

An enhanced-resolution grid system, presented in top view in Figure 11, was developed to provide a highly accurate computational grid model of the Space Shuttle Launch Vehicle. It includes 111 volume grids in an overlapped grid system that contains 16 million points. Each domain includes less than

500,000 points to allow Computational Fluid Dynamic simulations to be calculated on computers with moderate amounts of in-core memory. Most features larger than a four inch diameter have been included.

The ability to utilize the CAD interface for creating and modifying the geometry and for access to existing geometric databases proved to be very useful. It would have been inconceivable to consider creating a grid system of this complexity using previously available tools.

The Chimera grid interfacing scheme was useful in its flexibility both for decomposing the flow field geometry into separate domains and to divide the grid generation effort into separate autonomous tasks.

Given the current trend towards more accurate and complete representations of complex flow fields, it is important to maintain and enhance the tools suitable for modeling and manipulating large systems of geometry, data, and grids.

REFERENCES

1. Buning, P.G., Chiu, I.T., Martin, F.W., Jr., Meakin, R.L., Obayashi, S., Rizk, Y.M., Steger, J.L., and Yarrow, M., "Flowfield Simulation of the Space Shuttle Vehicle in Ascent," Proceedings of the Fourth International Conference on Supercomputing, April 1989.
2. Buning, P.G., Chiu, I.T., Obayashi, S., Rizk, Y.M., and Steger, J.L., "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent," AIAA-88-4359-CP, August 1988.
3. Martin, F.W., Jr., and Slotnick, J.P., "Flow Computations for the Space Shuttle in Ascent Mode Using Thin-Layer Navier-Stokes Equations," Applied Computational Aerodynamics (Progress in Astronautics and Aeronautics, Vol. 125), P.A. Henne, Ed., AIAA, Washington D.C., 1990, pp. 863-886.
4. Kandula, M., Pearce, D.G., Labbe, S.G., and Martin, F. W., Jr., "Effect of Solid Rocket Exhaust Plume on Space Shuttle Orbiter Flowfield," 19th JANNAF Exhaust Plume Technology Subcommittee Meeting, U.S. Army Missile Command, Redstone Arsenal, AL, May 1991.
5. Parks, S.J., Buning, P.G., Steger, J.L., and Chan, W.M., "Collar Grids for Intersecting Geometric Components Within the Chimera Overlapped Grid Scheme," AIAA-91-1587-CP, June 1991.
6. "ICEM Advanced Design Reference," Control Data Technical Publications, Pub. # 60000205 C, August 1991
7. "ICEM-CFD User's Guide Version 3.0," Control Data Technical Publications, August 1992
8. Buning, P.G., Chan, W.M., Renze, K.J., Sondak, D., Chiu, I.T., and Slotnick, J.P., "OVERFLOW User's Manual, Version 1.6p," NASA Ames Research Center, Moffett Field, CA, December 1991.
9. Steger, J.L., and Dougherty, F.C., "A Chimera Grid Scheme," Advances in Grid Generation, K.N. Ghia and U. Ghia, Eds., ASME FED Vol. 5, 1983, pp. 59-69.
10. Benek, J.A., Buning, P.G., and Steger, J.L., "A 3-D Grid Embedding Technique," AIAA-85-1523-CP, July 1985.
11. Benek, J.A., Steger, J.L., Dougherty, F.C., and Buning, P.G., "Chimera: A Grid-Embedding Technique," AEDC-TR-85-64, Arnold Engineering Development Center, Arnold AFS, TN, April 1986.
12. Benek, J.A., Donegan, T.L., and Suhs, N.E., "Extended Chimera Grid Embedding Scheme with Application to Viscous Flows," AIAA-87-1126-CP, July 1987.
13. "Space Shuttle Master Dimension Specifications," Rockwell International MD-V70, Rockwell Space Systems Division, Downey, CA, Revised 1991.
14. "Shuttle Vehicle Mold Lines and Protuberances", NASA Space Shuttle Interface Control Document Level II, ICD-2-00001 Rev K, NASA Johnson Space Center, Houston, TX, July 1991.
15. Chan, W.M. and Steger, J.L., "Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme," Appl. Math and Comp. Vol. 51, 1992, pp. 181-205.
16. Chiu, I.T., "UI User Manual," NASA Ames Research Center, 1992.
17. Walatka, P.P., Buning P.G., Pierce, L., Elson, P.A., "PLOT3D User's Manual Version 3.6," NASA Technical Memorandum 101067, NASA Ames Research Center, Moffett Field, CA, 1990
18. Suhs, N.E., and Tramel, R.W., "PEGSUS 4.0 User's Manual", AEDC-TR-91-8, Arnold Engineering Development Center, Arnold airforce Base, TN, June 1991.

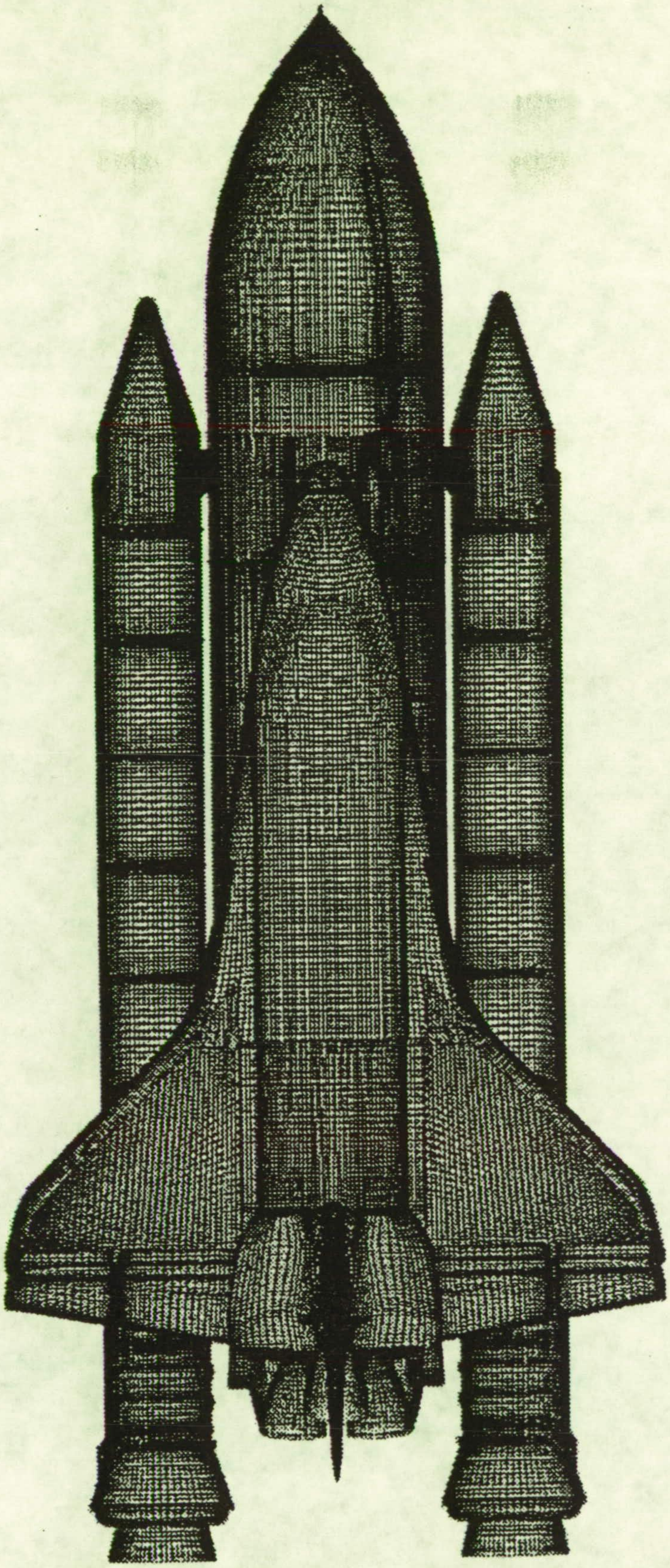


Figure 11. Top view of the surface grid for the enhanced-resolution Space Shuttle Launch Vehicle. Note the varied topology and density of the overlapped grids.