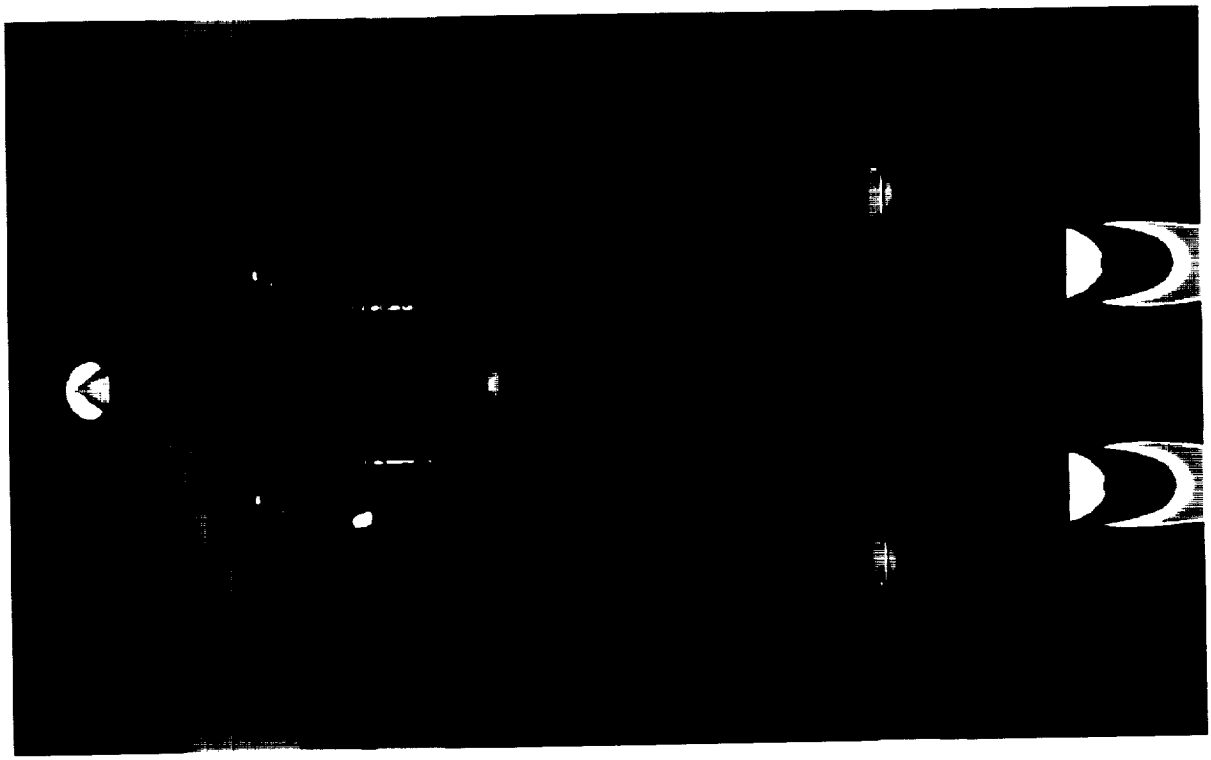
NASA-CR-197759

MCAT Institute
Final Report
95-10



# Development of Numerical Methods for Overset Grids with Applications for the Integrated Space Shuttle Vehicle

William M. Chan



January, 1995

NCC2-654

MCAT Institute
3933 Blue Gum Drive
San Jose, CA 95127

# DEVELOPMENT OF NUMERICAL METHODS FOR OVERSET GRIDS WITH APPLICATIONS FOR THE INTEGRATED SPACE SHUTTLE VEHICLE

**William M. Chan**
**MCAT Institute**
**NASA Ames Research Center**
**Mail Stop T27B-2**
**Moffett Field, CA 94035-1000**

## Abstract

Algorithms and computer code developments were performed for the overset grid approach to solving computational fluid dynamics problems. The techniques developed are applicable to compressible Navier-Stokes flow for any general complex configurations. The computer codes developed were tested on different complex configurations with the Space Shuttle launch vehicle configuration as the primary test bed. General, efficient and user-friendly codes were produced for grid generation, flow solution and force and moment computation.

## 1. Introduction

Computational fluid dynamics (CFD) analysis is widely used to simulate the flow field around complex multiple-body configurations, e.g. integrated Space Shuttle launch configuration,[1] V-22 tiltrotor,[2] Stratospheric Observatory For Infrared Astronomy (SOFIA)[3] and many others. In a structured grid approach, the complexity of the body usually requires the decomposition of the domain into a number of smaller blocks. An efficient method to use for this task is the Chimera overset grid scheme.[4] The first goal of this project is to enhance the capabilities of overset grid technology in the form of more robust and efficient algorithms and user-friendly computer codes. These include developments in the areas of grid generation, flow simulation and other pre-processing and post-processing utilities.

Overset grid methods have been effectively applied to the simulation of the time dependent flow over the integrated Space Shuttle vehicle during its ascent mode.[1] The second goal of this project is to assist the Shuttle group at NASA Johnson Space Center to obtain better comparisons between numerical simulations and flight data. With the accuracy of the numerical simulations verified, new simulations can then be performed to predict the aerodynamic behavior over a wide range of flight conditions.

---

The cover illustration shows the pressure coefficient plot for the Space Shuttle launch configuration with Mach number = 1.25, angle of attack = -5.1 degrees and Reynolds number = 270 million. The computation was performed by the shuttle group of NASA JSC. Computer software used include ICEM-CFD for surface grids, HYPGEN/HGUI for volume grids, BOXGR for Cartesian box grids, PEGASUS for intergrid connectivity and OVERFLOW for flow solution.

## 2. Grid Generation

Grid generation is typically the most time consuming step in CFD analysis. Some significant advances have been produced by this project in the form of more robust algorithms and user-friendly computer codes that reduce the user's effort required to generate high quality grids. Volume grid generation and the associated code HYPGEN are described in §2.1. Surface grid generation and the associated code SURGRD are described in §2.2. Other useful grid support tools are described in §2.3.

### 2.1. Hyperbolic Volume Grid Generator - HYPGEN

Volume grid generation for overset grids is most conveniently accomplished using hyperbolic methods. In addition to providing good grid point clustering and orthogonality control, hyperbolic methods are also one to two orders of magnitude faster than typical elliptic methods. Major enhancements to the robustness of the hyperbolic volume grid generation algorithm are described in Ref. 5. Earlier versions of the hyperbolic volume grid generation code HYPGEN are described in Ref. 6.

In October 1993, a major new version of HYPGEN and a major new version of its graphical user interface HGUI were released (version 2.0). The graphical user interface was developed by Dr. Ing-Tsau Chiu. A completely new set of examples with many aerospace applications was compiled. A more comprehensive user's manual was written for the combined HYPGEN/HGUI package. The final manuscript was published as a NASA TM.[7] The major new features of version 2.0 are described briefly below.

(1) A variable far field distance, a variable initial grid spacing and a variable end grid spacing can be specified in the input file *zetavar.i*.
(2) A user-defined stretching function in the normal direction can be specified in the input file *zetastr.i*.
(3) The input parameter list has been simplified. Only 4 parameters are now needed to specify the boundary conditions instead of 12 before.
(4) More robustness is introduced to the algorithm for treating constant planes boundary conditions by a metric rotation scheme.
(5) Algorithms were developed for some new boundary condition types: (a) floating collapsed edge with matching upper and lower sides for C and O-topologies, (b) constant interior planes in $x$, $y$ or $z$.

The HYPGEN/HGUI package is used by many different groups within NASA Ames and it has also been widely distributed to various groups outside of Ames. These include:

*Industry* - Boeing Commercial and Military, McDonnell Douglas Commercial and Aerospace, General Electric, Pratt and Whitney, Rockwell International, Northrop, Bell Helicopter, McDonnell Douglas Helicopter Company, Gulfstream Aerospace, Ford Motors Corp., Applied Research Associates,Inc., Nielson Engineering, Northern Research and Engineering, Sverdrup Technology Inc., Loral Aeronutronic, High Technology Corporation, Sparta, Inc., Bechtel Corporation, EER Systems Corp., Science Applications Internaional Corp., W. J. Schafer Associates, Inc.

*Government agencies* - NASA Johnson, Langley, Marshall, Lewis and Dryden, Sandia National Lab., Wright Lab., Eglin AFB, Department of the Navy, Naval Surface Warfare

Center, US Army Research Lab., Department of the Army.

*Universities* - Stanford University, Penn State University, University of California at Davis, University of California at Los Angeles, University of Southern California, Arizona State University, Naval Postgraduate School, Johns Hopkins University, University of Texas at Austin, Massachusetts Institute of Technology (Lincoln Laboratory), Carnegie Mellon University, Wichita State University.

*NGP* - Version 2.0 of HYPGEN has been incorporated as an option in the National Grid Project (NGP) headed by Prof. Joe Thompson at Mississippi State University.

## 2.2. Hyperbolic Surface Grid Generator - SURGRD

Although algebraic and elliptic methods can provide satisfactory surface grids for most applications, there are situations when it is more convenient to use hyperbolic methods. A good example is found in the generation of collar grids[8] in the Chimera overset grid scheme[4] for intersecting geometric components where there is no exact constraint on the location of three of the four grid boundaries. Furthermore, desirable grid attributes such as grid point clustering and orthogonality control are more easily attained using hyperbolic methods.

Surface domain decomposition and surface grid generation are the most time consuming steps in the overset grid generation process. Unlike the patched grid approach which requires neighboring grid boundaries to abut each other, the overset grid approach allows the surface grids to overlap each other. The implications of this freedom have not been fully utilized in most of the surface grid generation tools available today which are primarily tailored for the patched grid approach, e.g. GRIDGEN, EAGLEView, ICEMCFD. With the introduction of a hyperbolic surface grid generation tool, surface grid generation time can be reduced since only one edge of a four-sided surface grid needs to be specified. The edge that is chosen as the initial curve is typically some special control curve of the surface, e.g., a leading edge curve, an intersection curve between two components, a line along the symmetry plane, or a crease in the surface geometry. Thus there is some potential for automation in selecting the initial control curves. It must be emphasized that the scheme is not intended as a replacement for algebraic and elliptic methods but to serve as a useful and convenient alternative. The full advantages of the overset grid approach can be utilized by combining the scheme with algebraic or elliptic methods in an overset surface grid generator where the user can choose the appropriate scheme depending on the particular grid attributes desired.

Earlier developments of the hyperbolic surface grid generator SURGRD are described in Ref. 6. Recently, further significant enhancements have been made to SURGRD. These include methods to improve side and far field boundary control, a more robust projection scheme and extensions of the basic scheme to march over a collection of panel networks which implies that more complex geometric surfaces can now be treated. Each network consists of a rectangular array of panels (quadrilateral cells). Triangular cells are allowed only at the network boundary i.e., a singular collapsed point. Since the projection scheme is independent of the grid marching scheme, straight forward extensions can be made to grow the grids on a bicubic or NURBS representation of the reference surface instead of

the bilinear representation assumed in the present scheme. More details of the algorithm and code development for SURGRD are given in Ref. 9 and Appendix A.

### 2.3. Grid Support Tools - PROGRD, GRIDED, BOXGR

There are a number of other useful and essential tools that were developed to support the grid generation effort. These are described in detail in Ref. 10. A brief summary of each tool is given below.

The first is the PROGRD code that performs projection of a subset of an active surface grid on to a subset of a reference surface grid where bilinear representation of the reference grid is assumed. This procedure is needed in the overlapped regions between grids.

The second is the GRIDED code which is grid editor that has the capability to perform many operations that are common needed on a single grid in PLOT3D format, e.g. interchange J, K and L families, reverse grid indices, scale, translate, rotate, mirror, extraction of subsets, extrapolation of boundaries, concatenation into a second grid, etc. Before the existence of GRIDED, each of these tasks would require running a different code.

The third is the BOXGR code which is used to generate grids that enclosed the near field volume grids emanating from the body surface. Two types of grids are produced by BOXGR: The first type are layered and stretched Cartesian boxes that connect to the outer layers of volume grids from the body surface. These Cartesian boxes supply uniform resolution in the region between components. The second type is a stretched ellipsoidal grid that provides an efficient topology to connect the outer layers of the Cartesian boxes to the far field.

## 3. Overset Grid Flow Solver - OVERFLOW

Earlier developments of the 3-D compressible Navier-Stokes flow solver for overset grids, OVERFLOW,[11] are described in Ref. 6. Developments of OVERFLOW are primarily carried out by Dr. Pieter Buning. The author's contributions include the implementation of the Pulliam/Chaussee diagonal algorithm, the Jameson second and fourth order artificial dissipation, the force and moment computation routines and the symmetric Jacobian computation routine. The diagonal algorithm was adapted for overset grids by including *iblank* information and remains one of the most used options in OVERFLOW. More recent work by the author on OVERFLOW includes the investigation of a Cartesian solver option (§3.1) and the study and implementation of various central and upwind schemes for species convection (§3.2).

### 3.1. Investigation of Cartesian Option

The current grid generation strategy for complex configurations using the overset grid approach involves growing body conforming volume grids a short distance away from the solid body surfaces and then enclosing the volume grids in Cartesian box grids[12] (see also §2.3). These Cartesian box grids can take up a significant fraction of the total number of grid points (about 30% for the new grid system of the Space Shuttle Launch Configuration). In OVERFLOW, these Cartesian grids are treated like any other grids using generalized

coordinates. Therefore, an investigation was performed to estimate how many operations can be saved if subroutines using Cartesian operations are available in OVERFLOW. The results are summarized in Table 1. They indicate that only about a 20% savings in CPU time can be achieved using Cartesian subroutines for the box grids. It was concluded that the savings in CPU time is not significant enough to justify the development of Cartesian subroutines.

| RHS | Generalized | Cartesian | Percentage |
|---|---|---|---|
| Euler | 141 | 120 | 85% |
| Viscous | 282 | 195 | 69% |
| Smoothing | 357 | 333 | 93% |
| Norm | 17 | 17 | 100% |
| Delta $t$ | 6 | 6 | 100% |
| Total (viscous) | 803 | 671 | 83% |
| Total (inviscid) | 521 | 476 | 91% |

(a)

| LHS | Generalized | Cartesian | Percentage |
|---|---|---|---|
| Eigenvector | 292 | 131 | 45% |
| Euler | 105 | 84 | 80% |
| Viscous | 72 | 60 | 83% |
| Smoothing | 234 | 210 | 90% |
| Delta $t$/iden. | 8 | 8 | 100% |
| Matrix inv. | 114 | 114 | 100% |
| Total (viscous) | 825 | 607 | 74% |
| Total (inviscid) | 753 | 547 | 73% |

(b)

| MISC. | Generalized | Cartesian | Percentage |
|---|---|---|---|
| Delta $t$ scaling | 73 | 46 | 63% |
| Turb. model (B-L) | 105 | 78 | 74% |
| Laminar viscosity | 17 | 17 | 100% |
| Add delta $q$ | 11 | 11 | 100% |
| Total | 206 | 152 | 74% |

(c)

| TOTAL | Generalized | Cartesian | Percentage |
|---|---|---|---|
| viscous & turbulent | 1834 | 1430 | 78% |
| inviscid | 1480 | 1175 | 79% |

(d)

**Table 1.** Comparison of operation counts of diagonal scheme for generalized coordinates option versus Cartesian option. The percentage given is the ratio of the operation counts for the Cartesian option over the generalized coordinates option. (a) Right-hand-side operations. (b) Left-hand-side operations. (c) Miscellaneous operations (B-L stands for Baldwin Lomax turbulence model). (d) Total number of operations.

### 3.2. Species Convection

Several different gases are contained in the plumes produced by the Space Shuttle Main Engines and the Solid Rocket Boosters. It is assumed that diffusion of the plume gases is negligible and that there are no finite rate chemical reactions occuring. Hence, only convection will be modelled for the different gas species. A convection equation for the partial density of each species is solved. The Navier-Stokes equations and the convection equations are solved sequentially and weak coupling between the two sets of equations is obtained through the density and ratio of specific heats.

The present work involves investigating improvements that can be made to the numerical schemes used to solve the species convection equations in OVERFLOW. Two options were available in OVERFLOW to solve these equations: a basic implicit first order upwind scheme with no flux limiting and an implicit central differencing scheme with constant coefficient fourth order smoothing. Both of these schemes were not able to produce satisfactory results. The first order upwind scheme is too dissipative and the central differencing scheme with fourth order smoothing tends to produce undesirable overshoots. This prompted the investigation of three more schemes that can potentially provide some improvements:

(1) *Implicit second order upwind with no flux limiting*

This is just the second order version of the basic first order scheme already in OVERFLOW.

(2) *Implicit second order symmetric TVD*

A TVD scheme is chosen for its ability to capture sharp discontinuities. The second order symmetric TVD scheme of Yee-Roe-Davies[13] was selected since it is one of the cheapest TVD schemes in its class in terms of operation counts. A MUSCL scheme was also investigated but was abandoned due to difficulties in the linearization of the scheme for the implicit LHS of the equations.

(3) *Implicit second order central differencing with nonlinear second and fourth order smoothing*

The second and fourth order nonlinear artificial dissipation scheme of Jameson (same as the dissipation scheme in ARC3D) was modified for the species convection equation. The pressure sensor in the second order coefficient was replaced by a normalized second derivative of the species partial density. Since each species can have discontinuities in different regions of the flow, a different sensor for each species was necessary.

The 3-gas example in OVERFLOW was used to compare the above three schemes with the first order upwind scheme. In the example, the domain is a 2-D Cartesian grid and three gas species are introduced in the left inflow boundary of the grid. Gas one enters in the lower-third of the boundary, gas two enters in the middle-third of the boundary and gas three enters in the upper-third of the boundary. The entire domain is initially filled with gas one and the species are convected by a flow at 30 degrees to the lower boundary.

The partial density contours for gas two after 100 steps are shown in Figure 1 for the different schemes. Operation counts were also tallied for the three new schemes plus the first order upwind and central with fourth order smoothing schemes (see Table 2). The first order upwind is the cheapest but gave the most smeared result. The second order

upwind is the next cheapest and gave sharp discontinuities but the overshoots were quite significant. The TVD scheme gave fairly sharp profiles with no overshoot but is more expensive than the second order upwind. The central scheme is the most expensive with profiles similar to that of the TVD but has very slight overshoots.

| | Central-2/4 | Upwind-1 | Upwind-2 | TVD-sym | Central-4 |
|---|---|---|---|---|---|
| RHS | | | | | |
| convection | $(7+4N)N_d$ | $(22+5N)N_d$ | $(22+11N)N_d$ | $(7+17N)N_d$ | $(7+4N)N_d$ |
| smoothing | $(34+24N)N_d$ | 0 | 0 | 0 | $(49+11N)N_d$ |
| total | $(41+28N)N_d$ | $(22+5N)N_d$ | $(22+11N)N_d$ | $(7+17N)N_d$ | $(56+15N)N_d$ |
| LHS | | | | | |
| convection | $(8+2N)N_d$ | $27N_d$ | $35N_d$ | $(7+27N)N_d$ | $11N_d$ |
| smoothing | $(34+32N)N_d$ | 0 | 0 | 0 | $63N_d$ |
| delta t, I | $(8N)N_d$ | $6N_d$ | $8N_d$ | $(6N)N_d$ | $8N_d$ |
| pen/tri inv. | $(20N)N_d$ | $(7+5N)N_d$ | $(16+9N)N_d$ | $(9N)N_d$ | $(16+9N)N_d$ |
| total | $(42+62N)N_d$ | $(40+5N)N_d$ | $(59+9N)N_d$ | $(7+42N)N_d$ | $(98+9N)N_d$ |
| Grand Total | $(83+90N)N_d$ | $(62+10N)N_d$ | $(81+20N)N_d$ | $(14+59N)N_d$ | $(154+24N)N_d$ |
| Test Case CPU sec. | $7.56s$ | $5.94s$ | $6.08s$ | $7.09s$ | NA |

**Table 2.** Comparison of operation counts for various schemes. Central-2/4 = new central differencing scheme with 2nd and 4th order smoothing, Upwind-1, Upwind-2 = 1st and 2nd order upwind schemes, TVD-sym = symmetric TVD scheme, Central-4 = old central differencing scheme with 4th order smoothing, $N$ = number of species, $N_d$ = number of dimensions. The last row reports the CPU time in CRAY Y-MP seconds for 100 steps for the 3-gas example in the OVERFLOW package (NA=not available).

These results indicate that the TVD-sym scheme and the new Central-2/4 scheme can provide better results than the old schemes available in OVERFLOW. Further testing of these two new schemes were performed using an axisymmetric SRB (Solid Rocket Booster) case and an axisymmetric SSME (Space Shuttle Main Engine) case. In Figure 2, the old central scheme is compared with the TVD-sym scheme for the axisymmetric SRB plume. The TVD-sym scheme gives less overshoots and resolves more plume features. Not shown are the results for the new Central-2/4 scheme which are similar to that of the TVD-sym scheme. Although the TVD-sym scheme results in smaller overshoots, it is less robust than the Central-2/4 scheme, i.e. the TVD-sym scheme goes unstable more easily.

## 4. Force and Moment Computation

An important area of technology currently lacking in overset grid methods is that there is no accurate way to integrate the pressure and viscous stresses on solid surfaces to obtain force and moment information. This is because there is no simple scheme available to account for the overlapped zones on the surface grids. The work in this project involves the development of robust algorithms and computer codes to accurately compute the force and moment coefficients for a solid surface with proper accounting of overlapped regions.

This task is accomplished by blanking out points in the overlapped zones and then building triangular cells in the resulting gaps between neighboring grids. The pressure and viscous stresses can then be integrated accurately on the hybrid composite surface consisting of non-overlapping quads and triangles where the quads are just the unblanked cells in the original surface grids and the triangles are the newly generated triangular cells in the gaps.

Two modules are developed to perform the above task. The first module named MIXSUR is used to generate the hybrid composite surface and the second module named OVERINT is used to integrate the forces and moments on the composite surface. These modules are described in more detail below (further details on the algorithm are given in Appendix B).

## 4.1. Hybrid Composite Surface Grid Generation Module - MIXSUR

This module reads in a PLOT3D multiple grid file with iblanks and an input parameters file. The input parameters file is used to specify the subsets from the different grids that make up the solid surface where forces and moments are required The following are then performed:

(a) Automatically blank out points in the overlapped region between grids. For any pair of grids that overlap, grid points on the finer of the two grids in the overlapped region are retained. Grid points on the coarser of the two grids are blanked out. An integration iblank array is used to store this information.

(b) Automatically identify and order the points on the boundaries of the gaps created between grids due to the blanking operation.

(c) Automatically construct triangles in the gap between grids without introducing new points.

(d) Write out a multiple grid file with the integration iblank array and a multiple grid file containing information on the triangular cells.

The algorithms developed to perform steps (a), (b) and (c) are non-trivial and work is still in progress to improve the robustness of the algorithm such that a wide variety of case can be treated.

## 4.2. Force and Moment Integration Module - OVERINT

This module reads in a PLOT3D multiple grid file, a PLOT3D multiple grid solution file (OVERFLOW q file), an input parameters file plus the integration iblank array file and triangular cells connection file generated by MIXSUR. Integration of the pressure force and viscous force is performed on the hybrid composite surface consisting of quadrilateral and triangular cells. The user can input a reference area and length for the computation of the force and moment coefficients, and the x, y, z coordinates of the center of the axes about which moments are taken. Output from the module includes:

(a) the total area over which integration is performed,

(b) the lift, drag, side and the $X$, $Y$, $Z$ pressure and viscous force coefficients,

(c) the roll, pitch and yaw moment coefficients.

## 5. Applications

The software tools developed in this project are applicable to any general complex configurations. These tools are used by many different groups within NASA Ames and they are also widely distributed to industry and government laboratories (see §2.1). The primary test bed during the development of these codes is the integrated Space Shuttle launch configuration, e.g., grids are generated for the various Space Shuttle components using HYPGEN, the diagonal scheme implemented into OVERFLOW was tested on the old grid system for the launch configuration consisting of 9 grids (see Ref. 6 for more details). The work in this project was performed in collaboration with the Shuttle group at NASA Johnson Space Center. Extensive use of the grid generation and flow simulation software described in this report have been made by members in the Shuttle group at NASA JSC in performing launch vehicle calculations.[14,15,16]

In 1992, a major effort was taken to build a new grid system for the entire launch vehicle which included the modelling of many of the fine geometric features on the surface of the integrated vehicle. The development of the grids for the External Tank (ET), Solid Rocket Boosters (SRBs) and the Orbiter/ET Attached Hardware was carried out at NASA JSC. The development of the grids for Orbiter, the Cartesian box grids around the entire launch vehicle and the far field grid was performed by the author at NASA Ames. The final complete grid system consisted of 111 overset grids and 16 million points. An Orbiter alone flow solution was computed using OVERFLOW to test the robustness of the new Orbiter grids which included the modelling of the elevon gaps (see Ref. 10 for more details).

## 6. Concluding Remarks

Algorithms and computer software tools have been developed for the Chimera overset grid approach. Special attention was paid to the enhancement of robustness of the algorithms, improvement of efficiency (code optimization, increase of speed of execution), generality, user-friendliness of the software. Significant advances have been made in the area of hyperbolic surface and volume grid generation, flow solution computation, accurate force and moment computation, supporting software for pre-processing and post-processing of grid and solution. The recent work on SURGRD, version 2.0 of HYPGEN and preliminary developments on the force and moment integration modules MIXSUR and OVERINT were presented at the 2nd Overset Composite Grid and Solution Technology Symposium, October, 1994 (see Appendix C).

The computer codes developed are extensively used within NASA Ames and also have been widely distributed to U.S. industry, government laboratories and universities. The success of the software in treating very complex configurations has been demostrated in the accurate flowfield simulation of the integrated Space Shuttle launch vehicle using the new grid system at NASA JSC. Favorable agreement was obtained with wind tunnel and flight data.

## References


1. 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 Paper 88-4359-CP, August, 1988.

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

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

4. Benek, J. A., Buning, P. G. and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper 85-1523, July, 1985.

5. Chan, W. M. and Steger, J. L., "Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme," *Applied Mathematics and Computation*, Vol. 51, 1992, pp. 181–205.

6. Chan, W. M., "Numerical Methods for the Simulation of Complex Multi-body Flows with Applications for the Integrated Space Shuttle Vehicle," MCAT Institute Progress Report 92-009, 1992.

7. Chan, W. M., Chiu, I. T. and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, October, 1993.

8. 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 Paper 91-1587, *Proceedings of the AIAA 10th CFD Conference,* Honolulu, Hawaii, 1991.

9. Chan, W. M. and Buning, P. G., "A Hyperbolic Surface Grid Generation Scheme and its Applications," AIAA Paper 94-2208, AIAA 25th Fluid Dynamics Conference, Colorado Springs, CO, June, 1994.

10. Chan, W. M., "Numerical Computation of Complex Multi-body Compressible Navier-Stokes Flows with Applications for the Integrated Space Shuttle Vehicle," MCAT Institute Progress Report 93-05, 1993.

11. Buning, P. G., Chan, W. M., Renze, K. J., Sondak, D., Chiu, I. T. and Slotnick, J. P., "OVERFLOW User's Manual,"1994.

12. Pearce, D. G., Stanley, S. A., Martin, F. W., Gomez, R. J., Le Beau, G. J., Buning, P. G., Chan, W. M., Chiu, I. T., Wulf, A. and Akdag, V., "Development of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle," AIAA Paper 93-0533, 1993.

13. Yee, H. C., "A Class of High-Resolution Explicit and Implicit Shock-Capturing Methods," NASA TM 101088, February, 1989.

14. Gomez, R. J. and Ma, E. C., "Validation of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle," AIAA Paper 94-1859, *Proceedings of the AIAA 12th Applied Aerodynamics Conference,* Colorado Springs, CO, June, 1994.

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

16. Martin, F. W., Labbe, S., Pearce, D. G. and Wey, T. C., "Space Shuttle Launch Vehicle Wind Tunnel and Flight Aerodynamic Environments," AIAA Paper 94-1861, June, 1994.

## Appendix A

Chan, W. M. and Buning, P. G., "Surface Grid Generation Methods for Overset Grids," to appear in *Computers & Fluids*, 1995.

## Appendix B

Chan, W. M. and Buning, P. G. "Zipper Grids for Force and Moment Computation on Overset Grids," AIAA Paper 95-1681, to appear in Proceedings of the AIAA 12th Computational Fluid Dynamics Conference, San Diego, California, June, 1995.

## Appendix C

Chan, W. M., "Recent Developments in Grid Generation and Force Integration Technology for Overset Grids," 2nd Overset Composite Grid and Solution Technology Symposium, Fort Walton Beach, Florida, October, 1994.

# Appendix A

*Revised paper submitted to the journal Computers & Fluids*

# SURFACE GRID GENERATION METHODS FOR OVERSET GRIDS

WILLIAM M. CHAN *

MCAT Institute, NASA-Ames Research Center, Mail Stop T27B-2, Moffett Field, California 94035-1000

tel: (415) 604-6607, fax: (415) 604-1095, email: wchan@nas.nasa.gov

and

PIETER G. BUNING

NASA-Ames Research Center, Mail Stop T27B-2, Moffett Field, California 94035-1000

**Abstract**

Efficient methods for performing surface grid generation for overset (overlapping) grids are presented. These include a scheme using hyperbolic partial differential equations and an algebraic marching scheme. The combined use of hyperbolic and algebraic methods in easing the task of surface domain decomposition and surface grid generation for the Chimera overset grid approach is discussed. Extensions of a basic hyperbolic scheme are made to march over a collection of panel networks where each network consists of a rectangular array of panels (quadrilateral cells). Other enhancements include a more robust projection scheme and methods to improve grid boundary control. Examples are given for general applications in overset grid methods such as collar grids for intersecting geometric components and cap grids at fuselage noses and wing tips. More complex examples include surface grids for the Space Shuttle Orbiter and the V-22 tiltrotor.

* Author for correspondence.

*Revised paper submitted to the journal Computers & Fluids*

# SURFACE GRID GENERATION METHODS FOR OVERSET GRIDS

WILLIAM M. CHAN *

MCAT Institute, NASA-Ames Research Center, Mail Stop T27B-2, Moffett Field, California 94035-1000

tel: (415) 604-6607, fax: (415) 604-1095, email: wchan@nas.nasa.gov

and

PIETER G. BUNING

NASA-Ames Research Center, Mail Stop T27B-2, Moffett Field, California 94035-1000

**Abstract** - see attached sheet

## 1. INTRODUCTION

Computational fluid dynamics (CFD) analysis has become widely used in applications involving complex configurations [1-5]. In a structured grid approach to solving such problems, one of two different methods are commonly employed. The first is the patched grid method [6] where boundaries of adjacent grids are restricted to abut each other, and the second is the Chimera overset grid method [7] where the grid boundaries are not constrained as long as some overlap exists between adjacent grids (see Figure 1). As we will discuss below, the procedures for efficient grid generation using the two methods are quite different. While many algorithms and software tools have been developed for the generation of patched grids, the number of algorithms and software tools designed for overset grids is limited.

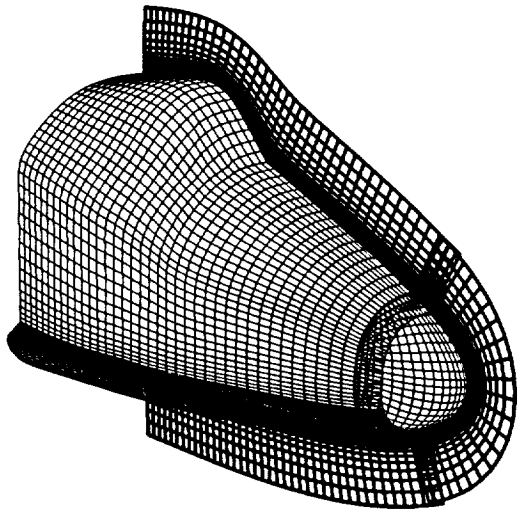


Fig. 1. Overset surface grids (thin lines) and volume grids (thick lines) for a fore-body geometry composing of a fuselage grid and a nose grid.

* Author for correspondence.

### *1.1. Overview of Grid Generation Process*

The grid generation procedure for overset grids can be broken into four main steps: (1) Surface domain decomposition, (2) Surface grid generation, (3) Volume grid generation and (4) Inter-grid connectivity. Typically, the first two steps combined can take up to 70% of the total grid generation time. With the introduction of more robust hyperbolic grid generation methods [8], volume grid generation typically requires less than 5% of the total time. The remainder of the time is spent on inter-grid connectivity.

It is worth examining the software tools that are available to the user to perform each of the steps. Surface domain decomposition involves partitioning the surface geometry into four-sided domains on which surface grids will be generated. Three-sided domains where one of the edges is collapsed to a point is allowed at the surface boundary. Very few software tools are available to assist the user in determining a choice of domain decomposition. Moreover, the time required to obtain a satisfactory domain decomposition depends heavily on the experience of the user. Many software tools are available for surface and volume grid generation using algebraic or elliptic methods, e.g. GRIDGEN [9], EAGLE [10], EAGLEView [11], ICEMCFD [12]. These are particularly suitable for the patched grid approach where all the boundaries of a grid have to be specified and constructed prior to the generation of the interior grid points. However, there are no general software packages available that are tailored for overset surface grid generation. Overset volume grid generation can be efficiently performed using tools based on hyperbolic methods such as the HYPGEN/HGUI package [13]. Software tools to perform inter-grid connectivity for complex configurations include the PEGSUS code [14], the DCF3D code [15], the BEGGAR code [16] and the CMPGRD code [17]. A more comprehensive list of grid generation software for overset grids can be found in Ref. [18].

### *1.2. Domain Decomposition*

For the patched grid approach, domain decomposition is a complex process since both the surface and the 3-D space around an object have to be decomposed into abutting blocks. For the overset grid approach, only the surface geometry needs to be decomposed in such a way that neighboring surface grids overlap. Volume grids will overlap naturally if sufficient overlap is provided on the surface.

Due to the lack of overset surface grid generation tools, overset grid users have been forced to employ patched grid tools to generate surface grids [1-5]. However, the overlap requirements for overset surface grids imply that not all four boundaries of the surface grid need to be precisely defined before generating the grid. A good strategy would be to first locate special control curves of the surface, e.g., a leading edge curve, an intersection curve between two components, a line along the symmetry plane, a curve along a slope discontinuity of the surface, or an open boundary of the entire surface. These curves can be used to assist the domain decomposition process since they can be employed as boundary curves for domains. Surface grids can then be grown by marching away from these control curves using hyperbolic or algebraic methods. Such a surface grid generator is described in this paper. Sometimes, a non-orthogonal grid with four-sided

control is required. The control curves can then be used as boundary curves for an algebraic or elliptic grid generator.

### *1.3. Surface Grid Generation*

Although algebraic and elliptic methods can provide satisfactory grids for most applications, there are situations when it is more convenient to use hyperbolic methods. For the overset grid approach, it is frequently the case that only one of the four boundaries of the surface grid needs to be specified and constructed prior to the generation of the interior grid. A good example is found in the generation of collar grids [19] for intersecting geometric components where the surface grid is generated hyperbolically by marching away from the intersection curve. Domain decomposition is simplified under the overset grid approach and the grid generation time with a hyperbolic scheme is relatively fast since only one boundary needs to be specified. Hence a significant reduction in the total grid generation time can be attained by the development of general and robust surface grid generation software tools tailored for the overset grid approach. The design cycle time for applications such as advanced subsonic and high speed civil transports can then be shortened as a result of the reduction in grid generation time.

The hyperbolic method of surface grid generation [20] involves marching a grid away from an initial boundary curve by a user-specified distance. This is achieved by the numerical solution of a set of hyperbolic partial differential equations. Desirable grid attributes such as grid point clustering and orthogonality control are naturally achieved. The predicted grid points are projected on to the underlying surface (reference surface) after each marching step. Different underlying surface definitions can be used, for example, a collection of panel networks (a collection of structured grids), a collection of triangular cells (unstructured grid), or a set of non-uniform rational B-spline patches (NURBS). Since the projection step is independent of the grid generation step, different types of reference surface definitions can be used without modifications to the grid marching scheme.

In the original scheme by Steger [20,21], the reference surface is a single panel network (single structured grid) and very limited side and far field boundary control is available. The possibility of covering a complex surface with overlapping hyperbolically generated surface grids was suggested by Steger [22]. However, at that time the capabilities of the algorithm and code were not general enough to perform such a task. In this paper, the methods used to enhance the generality and robustness of the overset surface grid generation code SURGRD are described. More complex surface geometry can now be accommodated by allowing a panel network collection (multiple structured grids) as the reference surface definition. A more robust projection algorithm and techniques for finer controls of the grid boundaries are developed and implemented. An algebraic marching scheme is included into the code for non-orthogonal grids.

The grid generation procedure has been formulated in physical space rather than parameter space. This is preferred since the physical space formulation provides direct grid spacing and orthogonality control. Moreover, working in parameter space becomes difficult when multiple panel networks are present in the

reference surface where the parameterization is usually discontinuous across network boundaries. Special treatment would also be needed for three-sided panel networks containing a singular point.

The contents of the paper are organized as follows. In §2, the governing equations for hyperbolic surface grid generation are given. The numerical procedures used to solve these equations and the algebraic marching scheme are described in §3. The various boundary conditions available for the hyperbolic option in the code are given in §4. Pre-processing that needs to be performed on the surface definition is discussed in §5. The projection algorithm used to ensure that the grid lies on the given reference surface is outlined in §6. In §7, applications of the current schemes to collar grids and cap grids are illustrated. Applications of the code on complex surfaces are illustrated by examples for the Space Shuttle Orbiter and V-22 tiltrotor in §8. Concluding remarks are given in §9.

## 2. HYPERBOLIC GOVERNING EQUATIONS

Consider generalized coordinates $\xi(x,y,z)$ and $\eta(x,y,z)$ where $\xi$ runs along some initial boundary curve and $\eta$ is the marching direction away from the curve on the reference surface. Also, let $\hat{n} = (\hat{n}_1, \hat{n}_2, \hat{n}_3)^T$ be the local unit normal that is assumed to be computable anywhere on the reference surface. The constraints of orthogonality of grid families and specified mesh cell area are

$$\vec{r}_\xi \cdot \vec{r}_\eta = 0, \tag{2.1}$$

$$\hat{n} \cdot (\vec{r}_\xi \times \vec{r}_\eta) = \Delta S, \tag{2.2}$$

where $\vec{r} = (x,y,z)^T$ is a position vector and $\Delta S$ is a user-specified surface mesh cell area. A third equation needed to close the system is provided by requiring that the marching direction of the grid be orthogonal to the surface normal at the local grid point, i.e., the marching direction is along the tangent plane of the underlying surface at the point (see Figure 2). This gives

$$\hat{n} \cdot \vec{r}_\eta = 0. \tag{2.3}$$

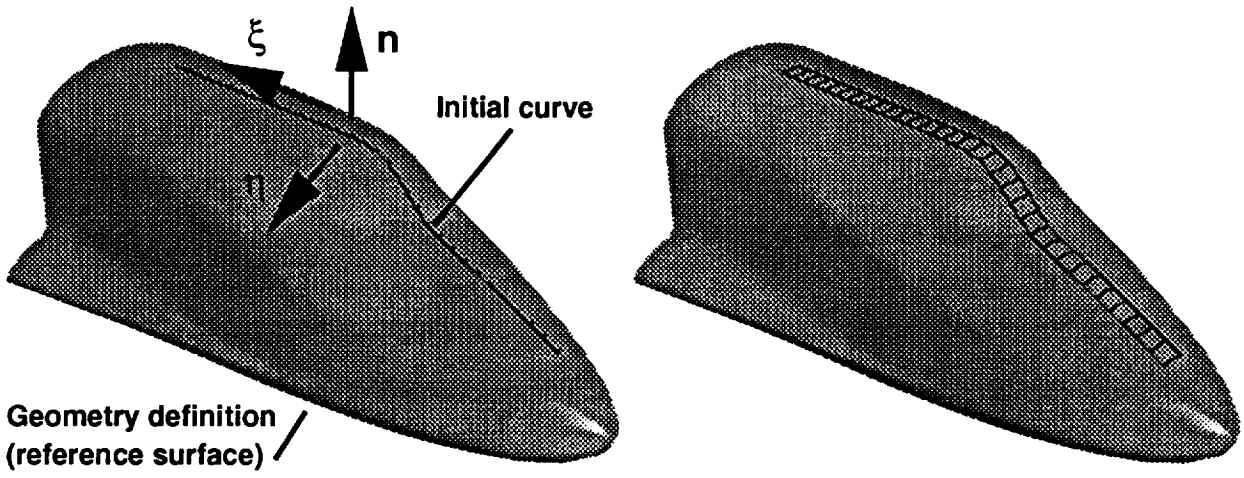
A unit vector in the marching direction $\eta$ can be obtained by the cross product of $\hat{n}$ with a unit vector in the initial curve direction $\xi$.

Equations (2.1)-(2.3) can be written out as

$$x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta = 0, \tag{2.4a}$$

$$\hat{n}_1(y_\xi z_\eta - z_\xi y_\eta) + \hat{n}_2(z_\xi x_\eta - x_\xi z_\eta) + \hat{n}_3(x_\xi y_\eta - y_\xi x_\eta) = \Delta S, \tag{2.4b}$$

$$\hat{n}_1 x_\eta + \hat{n}_2 y_\eta + \hat{n}_3 z_\eta = 0. \tag{2.4c}$$




(a) (b)

Fig. 2. (a) Diagram showing a reference surface and an initial curve where $\xi$ runs along the initial curve, $\eta$ is the marching direction along the local tangent plane of the reference surface and $n$ is the direction of the local unit normal of the reference surface. The unit vectors in the three directions $\xi$, $\eta$ and $n$ form a right handed set. (b) Surface grid obtained after marching one step from the initial curve.

In order to analyze Eqs. (2.4a-c), it is convenient to locally linearize about a known state $\vec{r}_0$. This can be written as

$$A_0\vec{r}_\xi + B_0\vec{r}_\eta = \vec{f}, \tag{2.5}$$

with

$$A = \begin{pmatrix} x_\eta & y_\eta & z_\eta \\ \hat{n}_3 y_\eta - \hat{n}_2 z_\eta & \hat{n}_1 z_\eta - \hat{n}_3 x_\eta & \hat{n}_2 x_\eta - \hat{n}_1 y_\eta \\ 0 & 0 & 0 \end{pmatrix} \tag{2.6a}$$

$$B = \begin{pmatrix} x_\xi & y_\xi & z_\xi \\ -\hat{n}_3 y_\xi + \hat{n}_2 z_\xi & -\hat{n}_1 z_\xi + \hat{n}_3 x_\xi & -\hat{n}_2 x_\xi + \hat{n}_1 y_\xi \\ \hat{n}_1 & \hat{n}_2 & \hat{n}_3 \end{pmatrix} \tag{2.6b}$$

$$\vec{f} = \begin{pmatrix} 0 \\ \Delta S + \Delta S_0 \\ 0 \end{pmatrix}. \tag{2.6c}$$

The equations in (2.5) have been shown to form a hyperbolic system for marching in the $\eta$ direction [21].

## 3. NUMERICAL MARCHING SCHEME

Marching schemes are used to perform surface grid generation in SURGRD. Two options are available: (1) the hyperbolic scheme for marching orthogonally from the initial boundary curve, and (2) the algebraic scheme for marching non-orthogonally from the initial boundary curve. A one-dimensional stretching function is used to prescribe the step size and marching distance in the marching direction. SURGRD employs a hyperbolic tangent stretching function [23] where the user can specify the number of grid points used, the

approximate distance to march out and the grid spacing at either one or both ends of the domain, i.e., at the initial curve boundary and at the far field boundary.

### *3.1. Hyperbolic Marching Scheme*

The nonlinear system of hyperbolic surface grid generation equations (2.4a-c) is solved with a finite-difference scheme. An unconditionally stable implicit scheme is used so that the marching step size in $\eta$ can be selected based only on considerations of accurately generating the grid. The nearby known state 0 is taken from the previous marching step. The product of the local arc length with the step size from the 1-D stretching function used to determine the input mesh cell area $\Delta S$.

Central differencing with explicit and implicit second order smoothing is employed in $\xi$ while a two-point backward implicit differencing is employed in $\eta$. The elements of $A$ contain derivatives in $\eta$. These derivatives can be expressed in terms of derivatives in $\xi$ using (2.4a-c) and they are computed as

$$(x_\eta, y_\eta, z_\eta)^T = B^{-1}\vec{g}, \tag{3.1}$$

where

$$B^{-1} = \frac{1}{\beta}\begin{pmatrix} x_\xi - \hat{n}_1 w & \hat{n}_2 z_\xi - \hat{n}_3 y_\xi & \hat{n}_1 s_\xi^2 - x_\xi w \\ y_\xi - \hat{n}_2 w & \hat{n}_3 x_\xi - \hat{n}_1 z_\xi & \hat{n}_2 s_\xi^2 - y_\xi w \\ z_\xi - \hat{n}_3 w & \hat{n}_1 y_\xi - \hat{n}_2 x_\xi & \hat{n}_3 s_\xi^2 - z_\xi w \end{pmatrix}, \tag{3.2a}$$

$$w = \hat{n}\cdot\vec{r}_\xi \quad = \hat{n}_1 x_\xi + \hat{n}_2 y_\xi + \hat{n}_3 z_\xi, \tag{3.2b}$$

$$s_\xi^2 = \vec{r}_\xi \cdot \vec{r}_\xi \quad = x_\xi^2 + y_\xi^2 + z_\xi^2, \tag{3.2c}$$

$$\beta = \mathrm{Det}(B) = s_\xi^2 - w^2, \tag{3.2d}$$

$$\vec{g} = (0, \Delta S, 0)^T. \tag{3.2e}$$

After putting the equations in delta form, the scheme can be written as

$$[I + B_{j,k}^{-1} A_{j,k}\delta_\xi - \epsilon_i(\Delta\nabla)_\xi]\,(\vec{r}_{j,k+1} - \vec{r}_{j,k}) = B_{j,k}^{-1}\,\vec{g}_{j,k+1} - [\epsilon_e(\Delta\nabla)_\xi]\,\vec{r}_{j,k}, \tag{3.3}$$

where

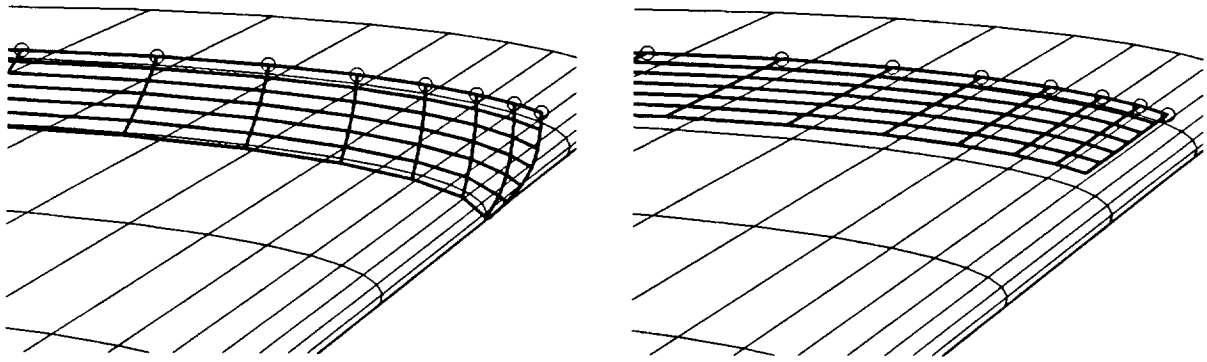
$$\delta_\xi \vec{r}_j = \frac{\vec{r}_{j+1} - \vec{r}_{j-1}}{2}, \quad (\Delta\nabla)_\xi \vec{r}_j = \vec{r}_{j+1} - 2\vec{r}_j + \vec{r}_{j-1}, \tag{3.4}$$

and $I$ is the identity matrix, $j$, $k$ are the grid indices in $\xi$ and $\eta$, respectively, $\epsilon_e$ and $\epsilon_i$ are the explicit and implicit smoothing coefficients, respectively with $\epsilon_i \approx 2\epsilon_e$. These can be chosen to vary spatially depending on geometric demands [8] or simply taken as constants. In (3.4), only the indices that change are indicated, i.e. $\vec{r}_{j+1} \equiv \vec{r}_{j+1,k}$, etc.

### *3.2. Algebraic Marching Scheme*

It is sometimes necessary to march away from an initial curve in a non-orthogonal direction where the hyperbolic marching scheme is not applicable (see Figure 3a,b). Frequently, the desired direction is along a family of curves defined by a panel network on the reference surface. The SURGRD code provides an algebraic marching option for this purpose.

Let the two families of curves defined by a panel network on the reference surface be labelled by $\xi_s$ and $\eta_s$, respectively. The particular panel network on which to march on and the direction in which the grid should grow are specified by the user. Four marching directions are possible: $\pm\xi_s$ and $\pm\eta_s$. For each point on the initial curve, the local $\xi_s$ and $\eta_s$ are determined (see §6). Suppose the grid is to be generated in the $+\xi_s$ direction. Then, each point on the initial curve is advanced in the $\xi_s$ direction along local $\eta_s$ = constant curves by an arc length using linear interpolation. The arc length used is given by the step size prescribed by the 1-D stretching function [23] in the marching direction. Since no smoothing mechanism is present in this scheme, it is intended to be used for smooth initial curves and that the total marching distance is assumed to be not large.



(a) (b)

Fig. 3. Comparison of hyperbolic and algebraic marching options (reference surface panels are represented by thin lines, points on initial curve are indicated by open circles, surface grid is represented by thick lines). (a) Hyperbolic marching results in a grid that is orthogonal to the initial curve. (b) Algebraic marching results in a grid that is parallel to a family of curves defined by the surface panels.

## 4. BOUNDARY TREATMENTS

For the hyperbolic option in SURGRD, boundary conditions are needed at the two boundaries emanating from the two ends of the initial curve. The different boundary condition types available in SURGRD are classified under the following three types.

(1) Periodicity - This is used when the initial curve is periodic.

(2) Floating - Under this class of boundary conditions, the grid points on the boundary can

  (a) float freely with no constraint,

  (b) float along an $x$, $y$ or $z$ =constant plane,

(c) float along a coordinate line on a specified network of the reference surface, or

(d) float along a user-specified 3-D curve defined by a set of points.

(3) Exact prescription - This is used when the exact locations of the grid points along the side boundaries are given by the user.

Boundary condition types 1, 2a, and 2b are treated implicitly in the solution of the governing equations. For boundary condition types 2c, 2d, and 3, the points on the boundary are predicted explicitly in advance and then a corrector step is applied to improve smoothness with the interior.

An important feature of the current approach is the ability to specify arbitrary boundary curves for the grid to follow (conditions 2c, 2d, and 3). The prescribed boundary is defined by an ordered string of points. For boundary conditions 2c and 2d, the predicted point on the boundary is made to lie on the linear segment between two consecutive points on the prescribed boundary curve. Since the prescribed boundary curve can be non-orthogonal to the initial curve, the smoothness of the grid near the boundary is maintained by allowing some non-orthogonality in the neighborhood of the prescribed boundary. This is achieved by rotating the $\eta$ derivatives in (2.6a) which represent the direction in which the grid marches. In the interior, this direction is normal to the previous $\eta$ = constant shell. Near the boundary, the $\eta$ derivatives are made to blend with the direction of the prescribed boundary curve where the amount of blending diminishes with distance from the boundary.

Further control of the stretching function in the marching direction is provided by allowing the far field distance and the initial and end grid spacing to vary for different points on the initial curve, i.e., the far field distance and grid spacings are functions of arc length along the initial curve. A convenient way to specify such a function is to supply the value of the function at specific node points (including the end points) on the initial curve. The value of the function for points between the node points are obtained by linear interpolation. Such a scheme is found to work well in SURGRD (see examples in §8).

## 5. PRE-PROCESSING OF SURFACE DEFINITION

The panel networks that define the surface geometry are allowed to be abutting, overlapping or have small gaps in between them. Four pre-processing steps have to be performed. The first step involves identifying deficiencies on the surface definition. It is the user's responsibility to fill in any large gaps between neighboring panel networks by supplying more networks or extrapolating the existing ones. Small gaps within a set tolerance level are acceptable for the current scheme. The user should also ensure that a sufficient number of points is provided on the panel networks to resolve regions of high surface curvature such as wing leading edges. Insufficient resolution can result in a faceted grid when bilinear interpolation is used.

The second pre-processing step involves checking the outward normals of the panel networks. The outward normal of a network is determined by the right-hand rule applied to the two families of grid lines defining the network. Normals from all the networks have to be made to point outwards from the surface.

An inconsistent normal direction for a network can be corrected by reversing the grid index of one of the grid families on the network.

The third pre-processing step is the identification of the neighbors of each panel network and it is automatically performed by the code. Each panel network is first labelled by a number. There are four distinct boundaries for a regular non-periodic network, three distinct boundaries for a network with a singular point at one end, and two distinct boundaries for a network with a periodic direction. For each point on the distinct boundaries of each network, three quantities $N_b$, $J_b$ and $K_b$ are identified and stored. The label number of the neighboring network is stored in $N_b$. If there is no neighboring network, the point is an outer boundary point of the entire domain and $N_b$ is set to zero. If there is a neighboring network, the $j$ and $k$ indices of the nearest cell on the neighbor are identified and stored in $J_b$ and $K_b$, respectively.

The algorithm used to identify $N_b$, $J_b$ and $K_b$ utilizes a 'minmax' box concept similar to that used by Chiu [24]. Each minmax box is defined by six numbers: the smallest and largest $x$, $y$ and $z$ coordinates of an object. A big minmax box is computed for the entire collection of points on each network and a small minmax box is computed for each cell of each network. For any point on the boundary of a panel network, the big minmax boxes are used to quickly eliminate networks that are not in the neighborhood of the point. The small minmax boxes are used to locate the cell (given by the $J_b$ and $K_b$ indices) on which the point lies. Tolerances are built into the code to allow for small gaps in between the panel networks.

The fourth pre-processing step is needed for the projection algorithm and is also automatically performed by the code. This involves generating a new point above each point of the surface networks by marching a small distance along the local surface normal. The new layer of points can be regarded as a phantom volume grid for the surface networks with a thickness of one cell. Surface normals for points on the boundary of a network are adjusted to blend with the surface normals of points on the corresponding boundary of a neighboring network. A point situated near the surface is then guaranteed to be within one of the cells or within the extrapolated volume of one or more cells in the phantom volume grid. Further discussions on the projection scheme are given in §6.

## 6. SEARCH AND PROJECTION ALGORITHM

After each grid marching step in $k$, the newly generated line of grid points are projected on to the reference surface. The projected points and the local surface normals at these points are used to perform the next grid marching step. For this paper, the reference surface is assumed to be a bilinear surface defined by a collection of panel networks. Each panel network consists of an ordered set of quadrilaterals. The projection is performed on to the set of bilinear surfaces formed by the quadrilaterals of all the networks. A singular point degeneracy is allowed on a network located on the outer boundary of the reference surface. For such a network, all the points at one boundary of the network are allowed to be coincident resulting in triangular cells rather than quadrilaterals.

For projection purposes, it is convenient to introduce the concept of families of networks which is described in §6.1. No projection is performed for points on the initial curve but the correct local surface normal must be selected in order to produce the desired marching direction. These issues are discussed in §6.2. The search and projection procedures performed after each marching step are outlined in §6.3.

### *6.1. Families of Panel Networks*

On the reference surface, a subset of the collection of networks that form a continuous surface is called a family, i.e., the entire collection of networks may consist of one or more families. This concept of a family allows the reference surface to contain networks that intersect each other. For example, at a wing/fuselage junction, the networks defining the wing are allowed to pierce into the networks defining the fuselage so that part of the wing is inside the fuselage. The wing networks belong to the wing family and the fuselage networks belong to the fuselage family. The user would then specify which family to project the current grid on to.

Consider an initial curve at the wing/fuselage intersection with points labelled such that $+\xi$ goes from trailing edge to leading edge on the upper surface of the wing. If the family concept is not invoked, then there are two valid directions to grow the grid: one along the wing into the inside of the fuselage and the other along the fuselage outside of the wing. Reversing the ordering of points (i.e. reversing the direction of $+\xi$) on the initial curve would result in two other possibilities being valid: one onto the wing outside the fuselage and the other on the fuselage but into the interior of the wing. Hence in order to avoid this ambiguity, a family needs to be specified to uniquely determine the marching and projection directions.

In general, points on the initial curve may not necessarily lie on the networks on which we wish to grow the grid on. For the wing/fuselage example above, the points on the initial curve may originate from and lie on the wing network. Note that these initial points are not projected and we can march a grid on to the fuselage side by specifying the fuselage family.

### *6.2. Initial Curves on Geometric Discontinuities*

Before the first marching step, the appropriate local surface normals for points on the initial curve are determined. For an initial curve lying on a smooth region of the reference surface, a unique local surface normal can be easily determined. However, for an initial curve lying along a slope discontinuity in the surface geometry (such as an airfoil trailing edge), the local surface normal on each side of the initial curve can be pointing in quite different directions. Special tests are performed to ensure that the correct normal is selected such that the desired marching direction is achieved by the right hand rule $\hat{r}_\xi \times \hat{r}_\eta = \hat{n}$ where $\hat{r}_\xi$ is a unit vector in the positive direction of the initial curve, $\hat{r}_\eta$ is a unit vector in the marching direction and $\hat{n}$ is the local unit normal. Given $\hat{r}_\xi$, the choice of one normal would result in $\hat{r}_\eta$ pointing to the interior of the reference surface (a valid marching direction) while choice of the other normal would result in $\hat{r}_\eta$ pointing outside the reference surface (an invalid marching direction).

### *6.3. Search and Projection after Each Marching Step*

After each marching step ($k > 1$), the new line of points are projected on to the bilinear representation of the reference surface and the local normals at the projected points are also determined by bilinear interpolation. Given a point $\vec{r_p}$ to be projected, the following steps are performed.

(1) For the $j$th point at level $k$, the initial guess for the network number $N_s$, the cell indices $J_s$ and $K_s$ are taken from the corresponding quantities at the $j$th point at level $k-1$.

(2) Local metrics based on the eight corners of the current cell in the phantom volume grid are constructed. Then a Newton iteration is performed to find the trilinear interpolation coefficients $\xi_c$, $\eta_c$ and $\zeta_c$ within the cell. Since we are interested in projection on to the surface, $\zeta_c$ is ignored when considering convergence and projection.

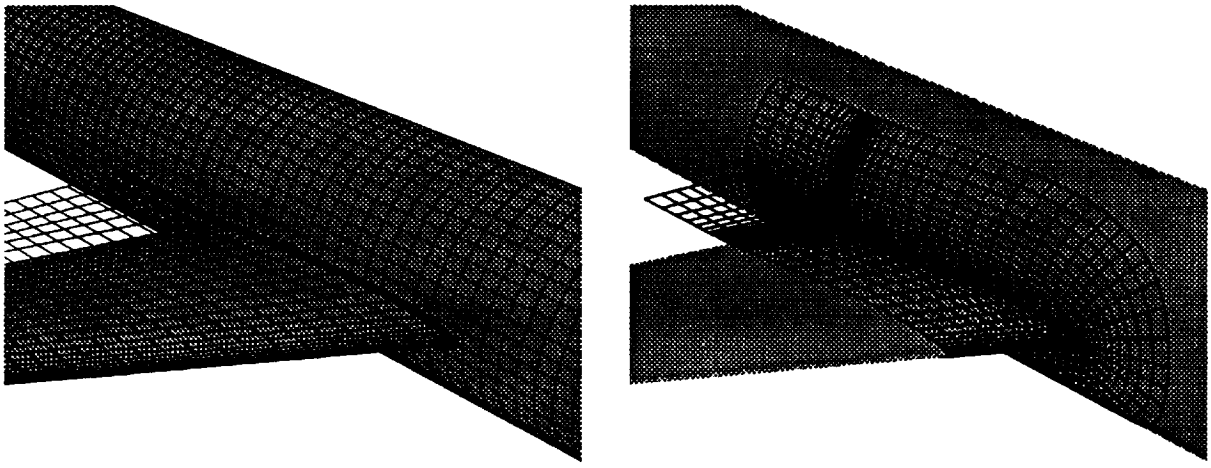
(3) A tolerance parameter $\epsilon$ (chosen to be about 0.005 or 0.5% of a cell dimension) is used in the convergence test. At convergence, if $-\epsilon < \xi_c < 1+\epsilon$ and $-\epsilon < \eta_c < 1+\epsilon$, then the point $\vec{r_p}$ must lie within this cell and the projected point is given by the bilinear coefficients $\xi_c$ and $\eta_c$ for the four points of the cell that lie on the surface. If either $\xi_c$ or $\eta_c$ is outside the above range, we move to the adjacent cell with direction given by $\xi_c$ and $\eta_c$. The current cell index $J_s$ is increased by one if $\xi_c > 1+\epsilon$ and decreased by one if $\xi_c < -\epsilon$. Similar movements for the current cell index $K_s$ are determined by $\eta_c$. If the current cell indices $J_s$, $K_s$ have not changed, then we have reached a boundary of the current network. Now we check if there is a neighboring network. If there is a neighboring network, we move to the appropriate cell in the neighboring network and go to step (2). If there is no neighboring network, the iteration is stopped since we have reached the boundary of the entire reference surface.

## 7. APPLICATIONS IN COLLARS AND CAPS

The primary applications of hyperbolic surface grid generation methods are currently found in the Chimera overset grid approach to gridding complex geometries. Two main types of grids are particularly convenient to generate using hyperbolic methods. The first type consists of the class of grids known as collar grids [19] while the second type is labelled as cap grids [4].

When two geometric components intersect, holes are created in the component grids by the overset grid approach and these holes overlap in the intersection region [19]. A collar grid serves as a convenient means to provide proper interpolation and resolution in such a region. Viscous grid clustering is needed in one direction only since the surface grid of the collar wraps around the intersection region. For example, a collar grid is employed at wing/body junctions (see Figure 4). The intersection line is used as the initial curve and surface grids can be grown on both sides of the initial curve using hyperbolic methods on the body side, and the algebraic marching method on the wing side. The two half-grids are then concatenated automatically in SURGRD to form the surface for the collar grid.

In order to close off the surface grid in geometric components such as fuselage noses and wing tips, grid singularities such as axis points or collapsed edges are quite often used. This can be detrimental to the flow



(a) (b)

Fig. 4. (a) Surface grids for wing and body. Initial curve for collar grid is indicated by thick black line. (b) Surface grid of collar for wing/body junction.

solution for two reasons. First, introduction of a singular line or plane in the field where grid points collapse at the grid boundary can result in jaggedness in the flow solution where it should be smooth. In the case of wing tips, these singularities occur in regions where the flow may have high gradients. Second, for many flow solvers, the maximum time step that can be taken is governed by numerical stability based on the maximum Courant number in the flow field. This maximum is inversely proportional to the smallest cell size of the grid. The smallest cell size for grids with singular axis points or collapsed edges can be quite small near the singularity.

The grid singularities mentioned above can be removed by the introduction of a rectangular patch or cap grid. A cap grid over the nose of the Space Shuttle Orbiter fuselage is shown in Figure 5. The upper surface of a cap grid for a rounded wing tip is shown in Figure 6a. Various slices of the volume grid for the cap grid are shown in Figure 6b.

## 8. APPLICATIONS IN COMPLEX CONFIGURATIONS

Figure 7 shows overlapping surface grids for the Space Shuttle Orbiter where the reference surface is a single network of rectangular cells (with triangular cells at the nose). All grids are generated hyperbolically except the two grids on either side of the crease line separating the Orbital Maneuvering System (OMS) pod and the main fuselage where the grids are generated by the algebraic option of SURGRD. The far field distance and the boundary control are specified by the user in a simple input file. A different far field distance is specified at two or three nodes and linear interpolation is used to produce a variable far field distance for most of the grids (see §4). If a grid is generated on each side of an initial curve such as at the wing fuselage junction and wing leading edge, the code automatically concatenates the grid from each side into a single grid. The system of 10 surface grids shown in Figure 7 (total of 10567 points) required 11 seconds of CPU time to generate on a Silicon Graphics Indigo 2 workstation.

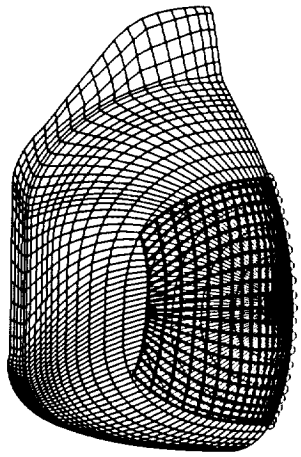
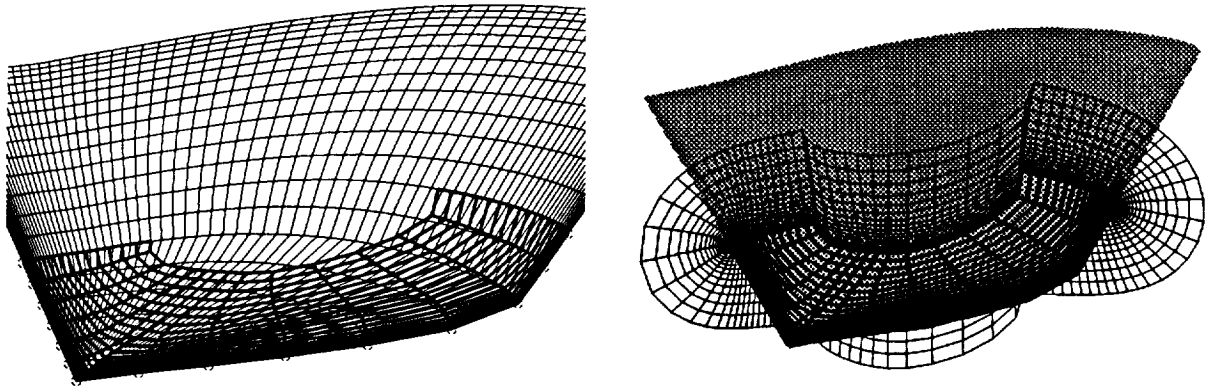


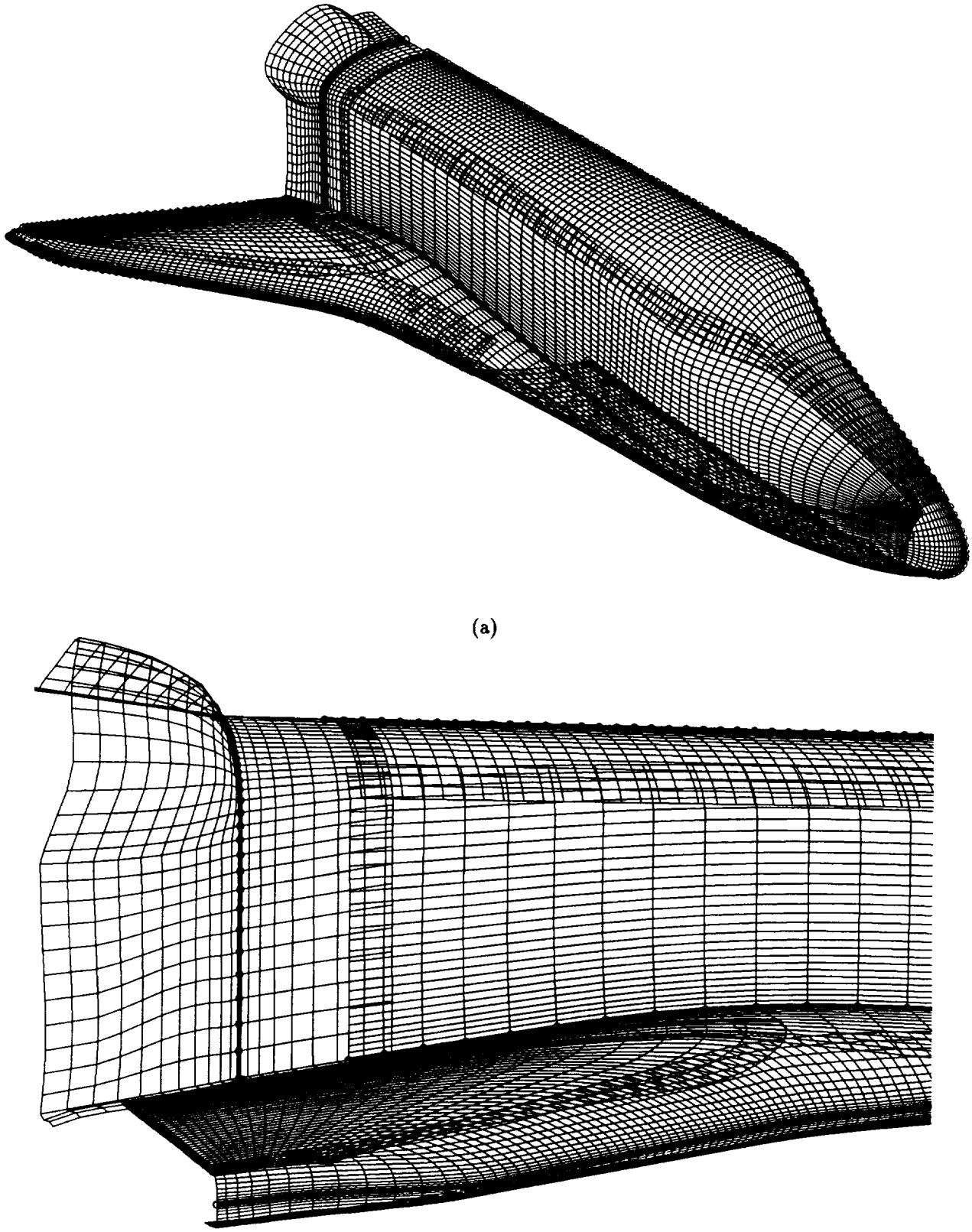
Fig. 5. Cap surface grid for the nose of the Orbiter fuselage (points on the initial curve are marked by open circles).



(a) (b)

Fig. 6. Cap grid for a wing tip. (a) Surface grid for the upper side of the wing is indicated by thin lines, surface grid for the upper side of the tip cap is indicated by thick lines, points on the initial curve are marked by open circles. (b) Various slices of the volume grid.

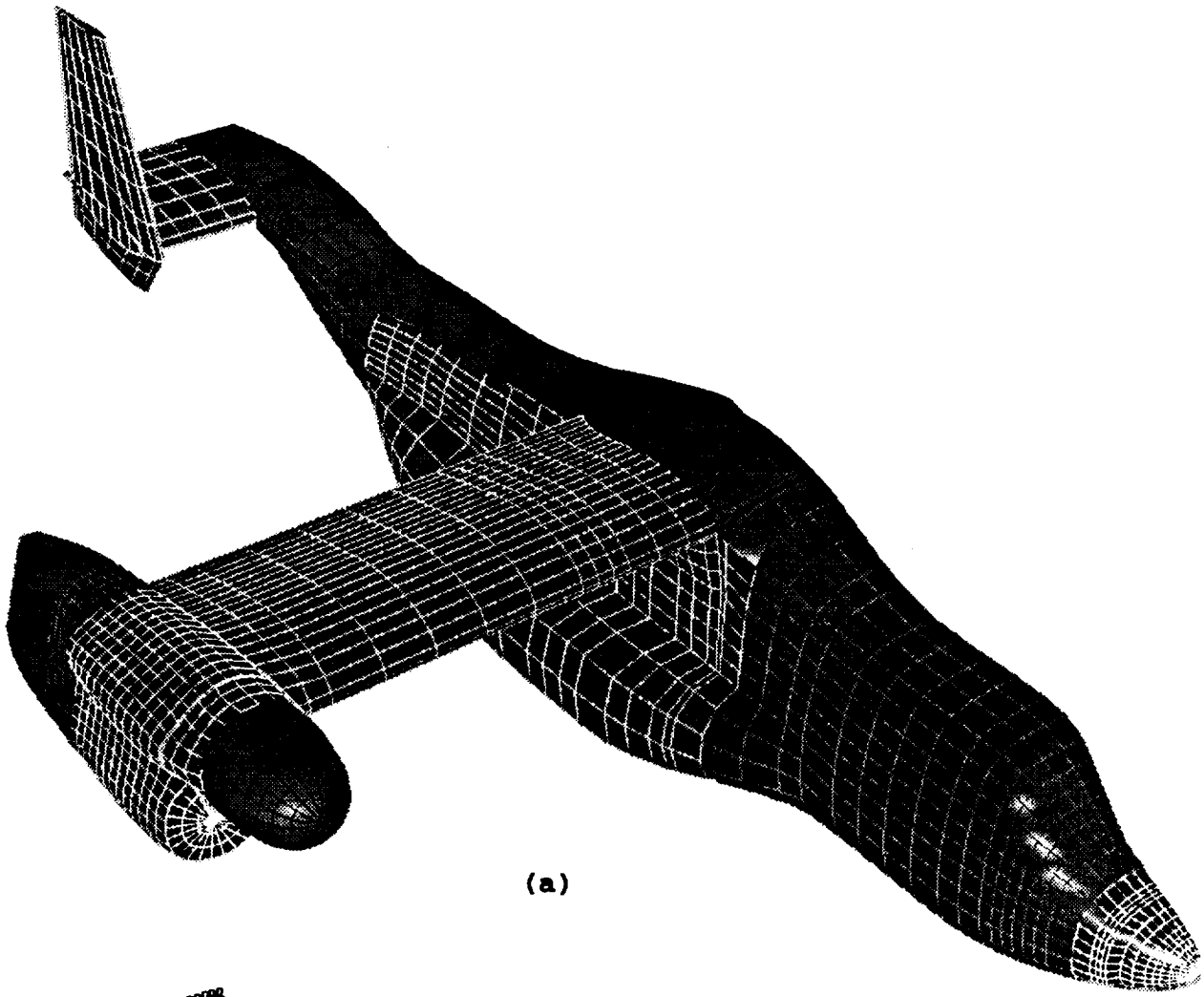
Figure 8a shows the surface definition for the right half of the V-22 tiltrotor [4] which consists of 22 panel networks. Most of the networks are abutting while some have gaps in between them. Initial curves are selected from subsets of the networks and grid points are redistributed on these initial curves. Overlapping surface grids are then grown on the 22-network reference surface. All grids are generated hyperbolically except for the wing part of the wing/fuselage collar and wing part of the wing/nacelle collar which are generated by the algebraic option. Overlapping surface grids for the fuselage, wing and nacelle are shown in Figure 8b. This system of 24 surface grids (total of 9441 points) required 9 seconds of CPU time to generate on a Silicon Graphics Indigo 2 workstation. The user's effort required to generate overset surface grids for the V-22 [4] could have been reduced significantly if a hyperbolic scheme such as the one described here was available.



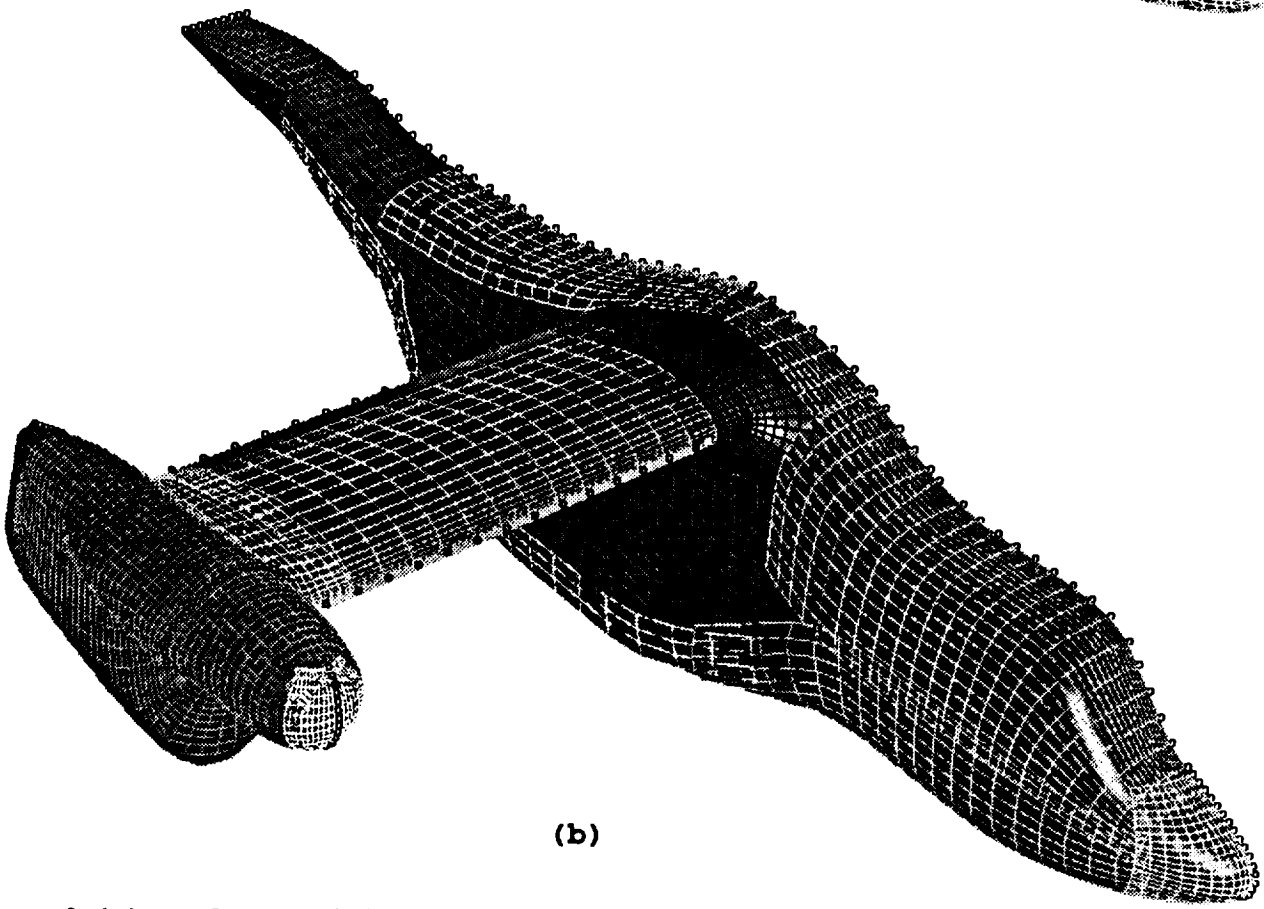
(a)

(b)

Fig. 7. Overlapping surface grids for the Space Shuttle Orbiter where points on the initial curves are indicated by circles joined by thick lines. (a) Front view, (b) back view.



(a)



(b)

Figure 8 (a) Surface definition for the right half of the V-22 tiltrotor consisting of 22 panel networks. (b) Overset surface grids for the fuselage, wing and nacelle where points on the initial curves are indicated by circles.

## 9. CONCLUSIONS

Overset surface grid generation methods, consisting of a hyperbolic surface grid generation scheme and an algebraic marching scheme, have been presented together with some of their applications. Surface grid generation time for overset grids is reduced since only one edge (instead of four that are required for typical algebraic and elliptic methods) needs to be defined for each surface grid. Extension of the scheme to admit multiple panel networks for the surface definition implies that more complex geometric surfaces can now be treated. Since the projection scheme is independent of the grid marching scheme, straight forward extensions can be made to grow the grids on a bicubic or NURBS representation of the reference surface instead of the bilinear representation assumed in the present scheme.

The SURGRD code offers a first step towards a surface grid generator tailored for overset grids which can potentially reduce the grid generation time by a significant amount. It must be emphasized that the hyperbolic scheme is not intended as a replacement for algebraic and elliptic methods but to serve as a useful and convenient alternative. The full advantages of the overset grid approach can be utilized by combining hyperbolic, algebraic and elliptic methods in an overset surface grid generator where the user can choose the appropriate scheme depending on the particular grid attributes desired.

A valuable addition to SURGRD would be a graphical interface to assist users in the selection of the initial curves. In the future, an algorithm can perhaps be developed to automatically construct the initial curves based on special control curves of the complex geometry.

## ACKNOWLEDGMENTS

The authors would like to thank Dr. Robert Meakin for his encouragement on this work, and to Dr. Ing-Tsau Chiu for some valuable discussions. The collection of panel networks for the V-22 tiltrotor was furnished by Bell Helicopter Textron, Inc. and Boeing Helicopters. This work is sponsored by the Space Shuttle Engineering Integration Office at NASA Johnson Space Center via NASA Ames Research Center under consortium number NCC2-654.

## REFERENCES


1. D. G. Pearce, S. A. Stanley, F. W. Martin, R. J. Gomez, G. J. Le Beau, P. G. Buning, W. .M. Chan, I. T. Chiu, A. Wulf and V. Akdag, Development of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle, AIAA Paper 93-0533 (1993).

2. R. J. Gomez and E. C. Ma, Validation of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle, AIAA Paper 94-1859, *Proceedings of the AIAA 12th Applied Aerodynamics Conference,* Colorado Springs, CO (1994).

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

4. R. Meakin, Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations, AIAA Paper 93-3350, *Proceedings of the AIAA 11th Computational Fluid Dynamics Conference*, Orlando, Florida, pp. 576–588 (1993).

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

6. M. M. Rai, A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations, *J. Comput. Phys.*, **62**, pp. 472–503 (1986).

7. J. A. Benek, P. G. Buning and J. L. Steger, A 3-D Chimera Grid Embedding Technique, AIAA Paper 85-1523 (1985).

8. W. M. Chan and J. L. Steger, Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme, *Appl. Math. & Comput.*, **51**, pp. 181–205 (1992).

9. J. P. Steinbrenner, J. .R. Chawner and C. L. Fouts, A Structured Approach to Interactive Multiple Block Grid Generation," *AGARD FDP Specialists Mtg. on Mesh Generation for Complex Three-Dimensional Configurations*, Loen, Norway (1989).

10. J. F. Thompson, Composite Grid Generation Code for General 3-D Regions - the EAGLE Code, *AIAA J.*, **26**, No. 3, pp. 271–272 (1988).

11. B. K. Soni, et al., GENIE$_{++}$, EAGLEView and TIGER: General and Special Purpose Graphically Interactive Grid Systems, AIAA Paper 92-0071 (1992).

12. V. Akdag and A. Wulf, Integrated Geometry and Grid Generation System for Complex Configurations, *Software Systems for Surface Modeling and Grid Generation*, Ed. Robert E. Smith, Hampton, VA, Langley Research Center, NASA CP 3143, pp. 161–171, (1992).

13. W. M. Chan, I. T. Chiu and P. G. Buning, User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface, NASA TM 108791, October (1993).

14. N. E. Suhs and R. W. Tramel, PEGSUS 4.0 User's Manual, AEDC-TR-91-8, AEDC/PA, Arnold Air Force Base, Tennessee (1991).

15. R. Meakin, A New Method for Establishing Intergrid Communication Among Systems of Overset Grids, AIAA Paper 91-1586, *Proceedings of the AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, Hawaii, pp. 662–671 (1991).

16. R. C. Maple and D. M. Belk, Automated Set Up of Blocked, Patched and Embedded Grids in the Beggar Flow Solver, *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, ed. N. P. Weatherill et al., Pine Ridge Press, pp. 305–314 (1994).

17. W. D. Henshaw, G. Chesshire and M. E. Henderson, On Constructing Three Dimensional Overlapping Grids with CMPGRD, *Software Systems for Surface Modeling and Grid Generation*, Ed. Robert E. Smith, Hampton, VA, Langley Research Center, NASA CP 3143, pp. 415–434 (1992).

18. Proceedings of the 2nd Overset Composite Grid and Solution Technology Symposium, Fort Walton Beach, Florida, Sponsored by Northwest Florida Section of AIAA (1994).

19. 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, *Proceedings of the AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, Hawaii, pp. 672–682 (1991).

20. J. L. Steger, Grid Generation with Hyperbolic Partial Differential Equations for Application to Complex Configurations, *Proceedings of the Third International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields.* Eds. A.S.-Arcilla, J. Häuser, P.R. Eiseman and J.F. Thompson, New York, North Holland, pp. 871–886 (1991).

21. 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, Univ. of Calif., Davis (1989).

22. J. L. Steger, Studies of the Chimera Method of Flow Simulation, Internal Report TM CFD/UCD 91-102-S, Department of Mechanical, Aeronautical and Materials Engineering, Univ. of Calif., Davis (1991).

23. M. Vinokur, On One-Dimensional Stretching Functions for Finite-Difference Calculations, *J. Comput. Phys.*, **50**, pp. 215–234 (1983).

24. I. T. Chiu and R. Meakin, On Automating Domain Connectivity for Overset Grids, AIAA Paper 95-0854 (1995).

# Appendix B

*Preliminary version of revised paper submitted to the AIAA 12th Computational Fluid Dynamics Conference, San Diego, California, June 19-22, 1995.*

# ZIPPER GRIDS FOR FORCE AND MOMENT COMPUTATION ON OVERSET GRIDS

William M. Chan * and Pieter G. Buning †
NASA-Ames Research Center
Mail Stop T27B-2
Moffett Field, California 94035-1000

## Abstract

A scheme is developed to account for the overlapped zones on overset structured surface grids when computing forces and moments for the surface. Grid points in the overlapped zone between grids are blanked. The resulting gaps are then filled by 'zipper grids' which are strings of triangular cells which conform to the original surface. A hybrid composite surface grid is then formed which consists of non-overlapping quadrilaterals and triangles, where the quadrilaterals are the unblanked cells in the original surface grids and the triangles are the newly generated triangular cells in the gaps. Pressure and viscous stresses which are known at the grid points can then be integrated accurately on the hybrid surface. Results obtained by applying the current scheme to numerical solutions are compared with data from wind tunnel tests for several complex configurations. These include the Space Shuttle Launch Vehicle, a subsonic transport and an oblique all wing configuration.

## 1. Introduction

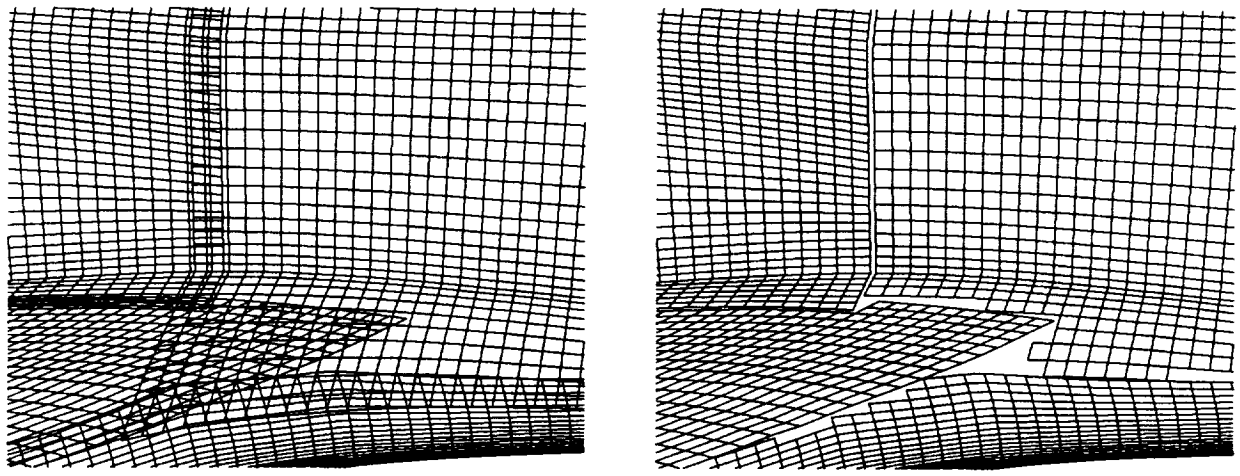
Chimera overset grid methods[1] have been successfully applied in the numerical flow simulations of complex configurations such as the Space Shuttle Launch Vehicle,[2,3,4] the V-22 tiltrotor,[5] the SOFIA Airborne Observatory,[6] and many others. Very good agreement of many flow characteristics has been obtained between the numerical solution and wind tunnel/flight tests. One of the most important set of quantities sought from numerical solutions is the set of force and moment coefficients for the solid surfaces in the flow field. Accurate computation of these quantities is crucial for aerodynamics design, optimization and moving body problems.

Forces and moments are usually computed by integrating the pressure and viscous stresses on the surface grids. Since the surface grids for overset grids are allowed to overlap arbitrarily, the regions in the overlapped zones are counted more than once if integration is performed over all the surface grids. In order for overset grid methods to be widely accepted for use in practical applications, a scheme must be available to accurately extract force and moment information from the numerical solution. A procedure developed to solve this problem is described in this paper.

The method for computing force and moment coefficients is broken into two main steps performed by two separate computer codes. The first step is to construct a hybrid composite surface grid consisting of non-overlapping quadrilaterals (quads) and triangles (the MIXSUR code), while the second step is to integrate the pressure and viscous stresses on the hybrid grid (the OVERINT code). Generation of the hybrid composite surface grid only needs to be performed once before the flow solver is called. The integration module OVERINT can be called just once at the end of the run to compute the final force and moment coefficients. It can also be called every $N$ number of iterations to supply the force and moment coefficients as indicators of convergence, or as input to a six-degrees-of-freedom rigid body dynamics package for moving body applications. In both cases, the connections between the quads and the triangles in the hybrid composite surface grid do not change with iterations.

Steps involved in the generation of the hybrid grid are described in §2. The integration scheme is presented in §3. In §4, the results of applying the current scheme on numerical solutions are compared with data obtained in wind tunnel tests, and results from using an alternative force/moment computation scheme. Summary and conclusions are given in §5.

* Research Scientist, MCAT Institute, Member AIAA.
† Research Scientist, NASA Ames Research Center, Senior Member AIAA



(a) (b)

**Figure 1.** (a) Surface grids for four overlapping subsets. (b) Surface grids after blanking of points in overlapped regions.

## 2. Hybrid Composite Grid Generation

An overset grid system consists of a collection of overlapping volume grids together with information on inter-grid connectivity. Let $J$, $K$ and $L$ be the grid indices of a typical volume grid. The surface domain on which forces and moments are to be computed can be described by a collection of surface subsets. Each surface subset is a subset of a $J$, $K$ or $L$ = constant slice of a volume grid in the system. For the rest of this paper, a surface subset will be referred to as a subset for short. An example with four overlapping subsets is shown in Figure 1a. A hybrid composite grid will be generated from the collection of subsets. This procedure consists of the following three main automated steps which are described in more detail in the subsections below.

(a) Points in the overlapped regions between subsets are blanked resulting in gaps between subsets. For any overlapping pair of subsets, only points from the coarser of the two subsets are blanked.

(b) Points on the boundaries of the gaps are identified and ordered. This creates a collection of strings of gap boundary points.

(c) Each pair of strings of gap boundary points are connected to form a zipper grid which is a single layer of triangular cells that fill the gap between neighboring subsets and conform to the original surface. In regions where more than two subsets overlap, multiple zipper grids are allowed to converge in a pocket of triangular cells.

An alternative to the above procedure is to retain all the points in all the subsets including those in the overlapped zones and then perform triangulation over the entire surface domain.[7] We have chosen not to do this for the following reason. If there are flow gradients not fully resolved by neighboring grids in the overlapped region, the solution will differ slightly. Use of points from all neighboring grids in this region would result in integration of an unsmooth pressure and/or viscous stress field. Although the hybrid grid generation procedure is more complex in the zipper grid approach, the effect of any solution mismatches is minimized by keeping as many non-overlapping quads as possible and introducing just a single layer of triangular cells. For conservation purposes, hybrid grids have also been used to replace overlapping structured grids in a 2-D or 3-D domain.[8] In that approach, all points in the overlapped region are blanked and the entire overlapped region is triangulated.

### 2.1. *Grid Point Blanking Scheme in Overlapped Zone*

The integration of forces and moments is performed on the set of non-overlapping quads and triangular cells. The set of non-overlapping quads is derived from the defined subsets where some of the points are blanked to ensure that the quads do not overlap. This information is stored in an integration iblank variable $B_{j,k}$ for each point on the subsets, where $B_{j,k}$ is defined by

$$B_{j,k} = \begin{cases} 1, & \text{if point is used for integration;} \\ 0, & \text{if point is not used for integration,} \end{cases} \tag{2.1}$$

and $j$, $k$ are grid indices for a subset. A quad is integrated if all four of its vertices have $B_{j,k} = 1$.

Initially, $B_{j,k}$ is set to the value 1 for all points on the subsets. Any point that is a hole point as a result of Chimera inter-grid connectivity[1] will have $B_{j,k}$ set to 0. Points in the overlapped region between subsets are then identified. For any pair of subsets that overlap, grid points on the coarser of the two subsets are blanked, i.e. $B_{j,k}$ set to 0. The coarser subset is defined to be the one with the fewer grid points in the overlapped region.

Points are identified to be inside an overlapped zone by the minmax box approach.[12] A minmax box is a collection of 6 numbers that indicate the smallest and largest $x$, $y$ and $z$ coordinates of the objects enclosed. A big minmax box is defined for each subset and small minmax boxes are defined for each quad of each subset. Then the blanking procedure goes as follows.

(1) Let $S_i$ and $S_j$ be any two subsets. Given a point in $S_i$, the big minmax box of $S_j$ is used to quickly determine whether the point falls inside of $S_j$. If it does, then the small minmax boxes of $S_j$ are used to determine whether the point falls inside any quad in $S_j$. If such a quad is found, then the point is counted as falling in the overlapped zone between $S_i$ and $S_j$.

(2) The number of points in $S_i$ that falls in the overlapped zone with $S_j$ is computed and stored in $N(S_i, S_j)$. Similarly, $N(S_j, S_i)$ is also computed.

(3) If $N(S_i, S_j) \leq N(S_j, S_i)$, then all the points in $S_i$ that falls in the overlapped zone with $S_j$ are blanked. This guarantees that all unblanked points in $S_i$ are not contained in any quad in $S_j$. However, the reverse is not necessarily true. Hence, a further test is necessary to blank any unblanked points of $S_j$ that falls inside an unblanked quad in $S_i$. The procedure follows similarly with the role of $S_i$ and $S_j$ reversed if $N(S_j, S_i) < N(S_i, S_j)$.

(4) This is performed for all pairs of subsets.

(5) After all the blanking is completed, some loose strings of unblanked points may remain which are surrounded by blanked points. A final test is thus performed to blank out these loose strings of points.

Figure 1b shows the remaining non-overlapping quads after the blanking step is performed on the example in Figure 1a. Since the surface grid lines are generally not aligned with the Cartesian coordinates axes, the use of minmax boxes will tend to blank out more points than is necessary. A more accurate but also more expensive test can be used to reduce the size of the gaps created.

### 2.2. *Stringing of Gap Boundary Points*

The second step in the hybrid grid generation procedure is to identify and order the points on the gap boundaries created by the blanking step. These points fall under two types: (I) points on the boundary of holes defined by the integration iblank variable, and (II) points on the boundary of subsets that lie in the overlapped region with another subset.

Type I points are identified by examining $B_{j,k}$. First, it must be an unblanked point. For an interior point $(j, k)$ to qualify as a Type I point, at least 3 and no more than 7 of its 8 neighboring points must be unblanked. The neighboring points are located at $(j-1, k-1)$, $(j-1, k)$, $(j+1, k)$, $(j, k-1)$, $(j, k+1)$, $(j-1, k+1)$, $(j, k+1)$ and $(j+1, k+1)$. For a point on the boundary or a corner of a subset, it is a Type I point if any one of its 3 or 2 neighbors are blanked, respectively. Type II points are identified by testing if any boundary points on a given subset lie inside any small minmax box of a cell from another subset.

After identifying the gap boundary points, they need to be connected to form ordered strings of points. Two gap boundary points are connected if an edge of a subset exists between them. The connection of the gap boundary points then results in a collection of gap boundary strings (see Figure 1c).

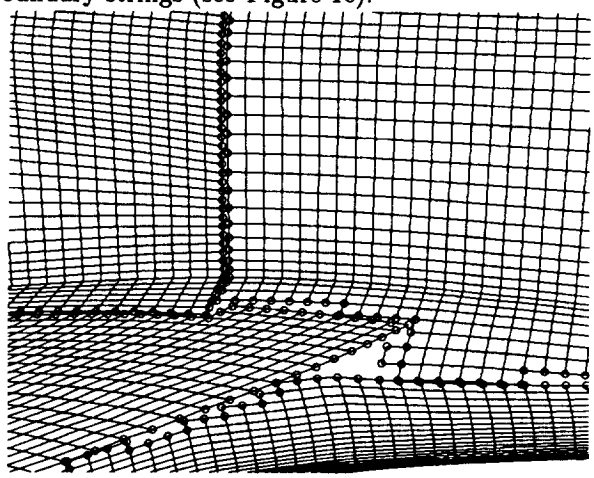
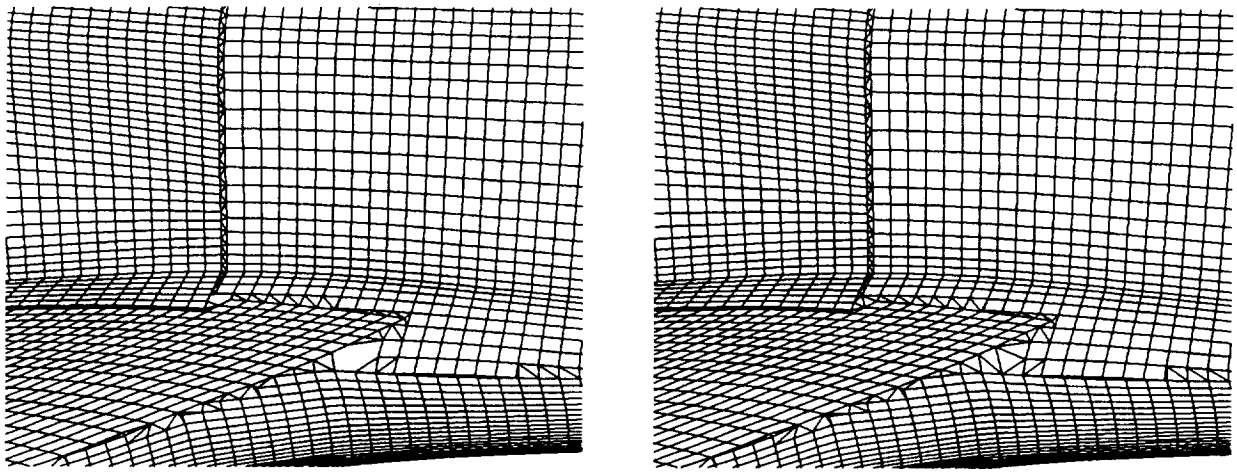


**Figure 1.** (c) Points on gap boundary strings are indicated by circles joined by thick lines.



(d) (e)

**Figure 1.** (d) Gaps between pairs of boundary segments are filled by zipper grids where each consists of a single layer of triangular cells. (e) Final hybrid composite surface grid with remaining polygonal gaps filled with triangles.

### 2.3. *Construction of Zipper Grids*

Triangles are constructed by connecting points between pairs of gap boundary strings resulting in a set of zipper grids. Each zipper grid is a single layer of triangular cells. No new points are introduced in the generation of these cells which are required to conform to the original surface geometry. The steps involved in this procedure are described below.

#### 2.3.1. *Boundary Segment Matching*

A zipper grid is formed by connecting points between segments of two boundary strings. Each boundary string $R_i$ is divided into segments as follows. For each point $P_i$ on $R_i$, the nearest point on another boundary string $R_j$ is located. The point $P_i$ is now mapped to the boundary string $R_j$. A segment of $R_i$ is a set of contiguous points on $R_i$ that is mapped to $R_j$. A segment from $R_i$ that is mapped to $R_j$ is matched with a segment from $R_j$ that is mapped to $R_i$.

#### 2.3.2. *Zipping*

The matching step identifies segment pairs that should be connected via a zipper grid. Let A and B be a segment pair to be connected. The algorithm used to construct a zipper grid is as follows.

(a) *Selection of initial points*

The first connection between A and B is formed by searching for a starting point in A and a starting point in B to connect to each other. For a non-periodic segment pair A and B, the two end points of segment A and B that are closest to each other are chosen as the starting points of the zipping process. For a periodic segment pair A and B, any two closest points between the two segments can be used as starting points.

(b) *Connection to form triangles*

Let $P_i$ be the $i$th point on segment A and $P_j$ be the $j$th point on segment B where $i = j = 1$ at the starting points. Also let $i_{\max}$, and $j_{\max}$ be the number of points on A and B, respectively. For $i < i_{\max}$ and $j < j_{\max}$, $P_i$ may be connected to $P_{j+1}$ or $P_j$ may be connected to $P_{i+1}$. The first step is to test if the quad formed by the points $P_i$, $P_{i+1}$, $P_{j+1}$ and $P_j$ is convex. The connection of $P_i$ to $P_{j+1}$ splits the quad into two triangles. If the normals of these triangles have the same sign, then the connection is valid. A similar test is performed for the $P_j$ to $P_{i+1}$ connection. In cases where only one of the two connections is valid, the valid connection is chosen. In cases where both connections are valid, the quad is convex and the connection with the shortest distance is chosen. However, if both connections are invalid, the quad is crossed inside out. The shortest distance test is still performed and can result in good triangulations in some situations. When bad triangulations are obtained, we will proceed to the next connection but the bad triangles are flagged. When $i = i_{\max}$ is reached (last point of segment A), all remaining points in segment B are connected to the last point in segment A, and vice versa.

(c) *Reconnection to remove bad triangles*

When there is a large discrepancy between the grid resolutions of the neighboring grids in the overlapped zone, and when the grid lines are not aligned, the zipping procedure described above can sometimes produce bad triangles. These triangles are bad in the sense that they overlap some existing quad or good triangle. From experience, typically only less than 1% of the triangles created are bad under the conditions stated above. Some of these bad triangles can be removed and good triangles can be formed by performing the zipping step (b) in the reverse direction in the region of bad triangles. Zipper grids for the example started in §2.1 are shown in Figure 1d.

### 2.3.3. *Closing of Pockets*

After connecting pairs of gap boundary segments, most of the gaps created by the blanking procedure described in §2.1 are now closed by triangular cells. However, there may be small gaps left in regions where three or more subsets overlap. Each small gap is in the form of a closed polygon where each edge of the boundary is either an unconnected edge of a gap boundary segment or an end edge of one of the zipper grids. These edges are identified and connected to form a closed loop. The gap inside the closed loop is then triangulated without introducing new points using a similar zipping algorithm as described in the previous subsection. First, a starting edge has to be chosen. This is selected to be the edge whose end points contain interior angles that are closest to 90 degress. Zipping then proceeds from the starting edge until the entire polygon is filled with triangles. The complete hybrid composite grid for the example started in §2.1 is shown in Figure 1e.

## 3. Pressure and Viscous Stress Integration

Integration of the pressure and viscous stresses is performed on the hybrid composite surface consisting of a collection of non-overlapping quadrilaterals and triangles. For each quadrilateral or triangle, the value of the function being integrated is known at the vertices. A number of quadrature rules can be used to perform the integration. For simplicity, the function is assumed to be a constant inside the area of the quadrilateral or triangle where the constant is the arithmetic average of the values of the function at the vertices. This can be written as

$$\int_{polygon} f\, d\vec{A} \approx \left( \frac{1}{N_v} \sum_{i=1}^{Nv} f_i \right) \cdot \vec{n} \Delta A, \qquad (3.1)$$

where $f$ is a scalar function to be integrated, *polygon* is either a quadrilateral or a triangle, $N_v$ is the number of vertices of the polygon, $f_i$ is the value of the scalar function at the vertices, $\vec{n}$ is the unit normal of the polygon, and $\Delta A$ is the area of the polygon. A similar expression can be written for the integral of a vector function $\vec{f}$. For a triangle, the normal is uniquely defined. For a quad, the normal is computed by the cross product of the diagonals. The integral in (3.1) is performed over all the unblanked quads and triangles and the results are summed and normalized to obtain the force and moment coefficients.

There are three possible ways to perform the quadrature over a quad by treating it as (1) a 4-point polygon, (2) two triangles by connecting one diagonal of the quad, (3) two triangles by connecting the other diagonal of the quad. Let $I_q$, $I_{t1}$ and $I_{t2}$ be the value of the integral computed using options (1), (2) and (3), respectively. It can be shown analytically that if the quadrature rule given by (3.1) is used on a parallelogram, then

$$I_{t1} \leq I_q \leq I_{t2}. \qquad (3.1)$$

Hence, for a quad that is a parallelogram, option (1) is the better choice since options (2) and (3) tend to produce undesirable biases. However, as a quad degenerates to a triangle as in the case of two adjacent points becoming very close together, option (2) or (3) is more attractive since the two close points are not counted twice. On typical surface grids for CFD applications, most of the quads are approximately parallelograms in shape. We have therefore chosen option (1) for the quadrature on quads. Cells adjacent to a singular axis point on a surface subset are integrated as triangles.

## 4. Comparison with Wind Tunnel Data

The scheme described above is used to compute the force and moment coefficients for a number of CFD solutions on complex configurations obtained using overset grids. These include the Space Shuttle Launch Vehicle, a subsonic transport configuration and an oblique all wing configuration. The results are compared with data from wind tunnel tests. In the case of the Space Shuttle Launch Vehicle, results from another independent force and moment computation scheme are also available for comparison.

### 4.1. *Space Shuttle Launch Vehicle*

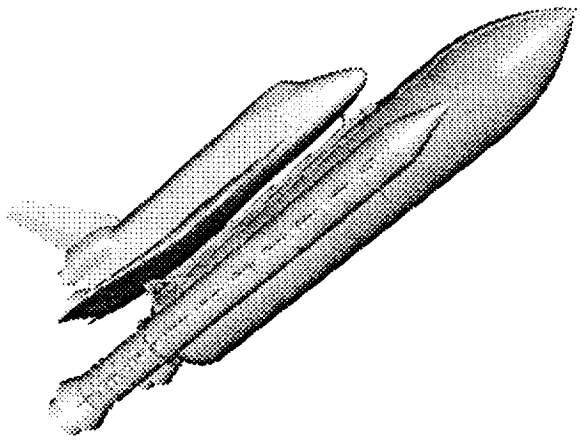


**Figure 2.** Components of the Space Shuttle Launch Vehicle.

The highly complex configuration of the Space Shuttle Launch Vehicle[2,3,4] (see Figure 2) offers a good test bed for the scheme described in this paper. The overset grid system contains 113 grids and 16.4 million grid points. Numerical solutions were computed by the Ascent Wing Aerodynamic Loads Verification Team at NASA Johnson Space Center (JSC) using the Chimera Navier-Stokes flow solver OVERFLOW.[10] In addition, wind tunnel measurements are available where the data is compiled and recorded in a data base - the Operational Aerodynamic Databook (OADB) database.[9] Although measurements were carried out under wind tunnel Reynolds numbers, the data has been adjusted using flight measurements to allow predictions for flight Reynolds numbers. This data is compared with the coefficients obtained using the scheme described in this paper applied to a CFD solution at a Mach number of 1.05, angle of attack of -3.08 degrees, and a Reynolds number of 306 million. As a further check of the current scheme, the results are also compared with the coefficients obtained by a different integration program developed by Dr. Thomas Wey at Lockheed Engineering under consortium with NASA JSC.[4] The JSC program triangulates the entire vehicle and performs integration on the resulting triangles.

The following notation is used in Tables 1, 2, 3a and 3b in the subsections below.

Databook = value from Operational Aerodynamic Databook database with uncertainties,
CFD-JSC = CFD solution with independent integration procedure at NASA JSC,
CFD-ZIP = CFD solution with current zipper grid procedure.

$C_a$ = Axial force coefficient,
$C_s$ = Side force coefficient,
$C_n$ = Normal force coefficient,
$C_{mr}$ = Rolling moment coefficient,
$C_{mp}$ = Pitching moment coefficient,
$C_{my}$ = Yawing moment coefficient.

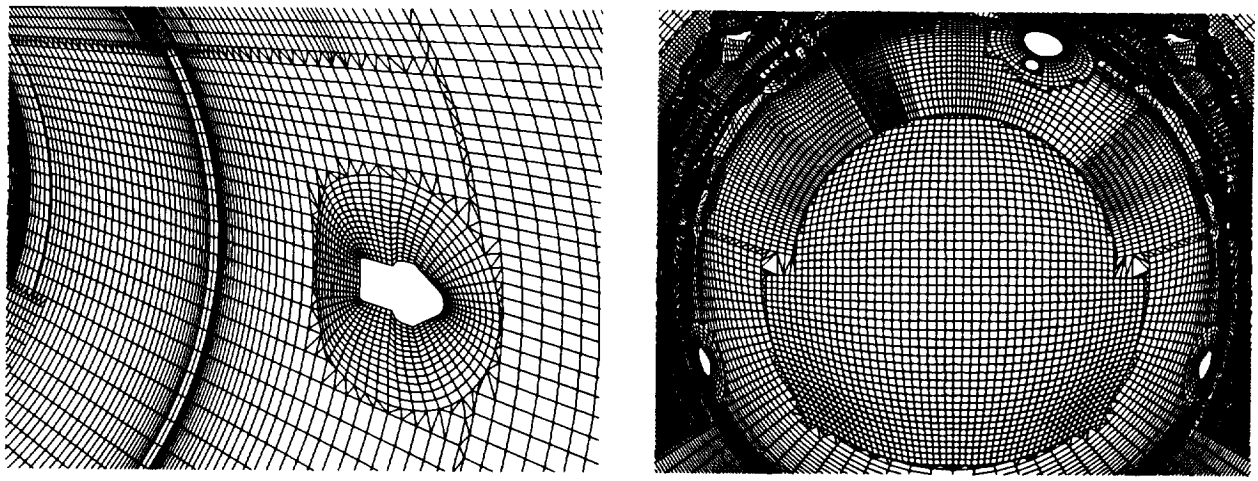
Results for the External Tank, Orbiter and Solid Rocket Boosters are given in the subsections below.

#### 4.1.1. *External Tank*

The first comparison is performed on the External Tank (ET). Zipper grids generated in the forward attach hardware region and the base region of the ET are shown in Figures 3a and 3b. Comparison of force and moment coefficients with wind tunnel data is given in Table 1. The discrepancy in the axial force coefficient is due to the fact that the Databook and CFD-JSC values include the blockage effects of the Orbiter/ET attach hardware while the CFD-ZIP value does not include such effects.

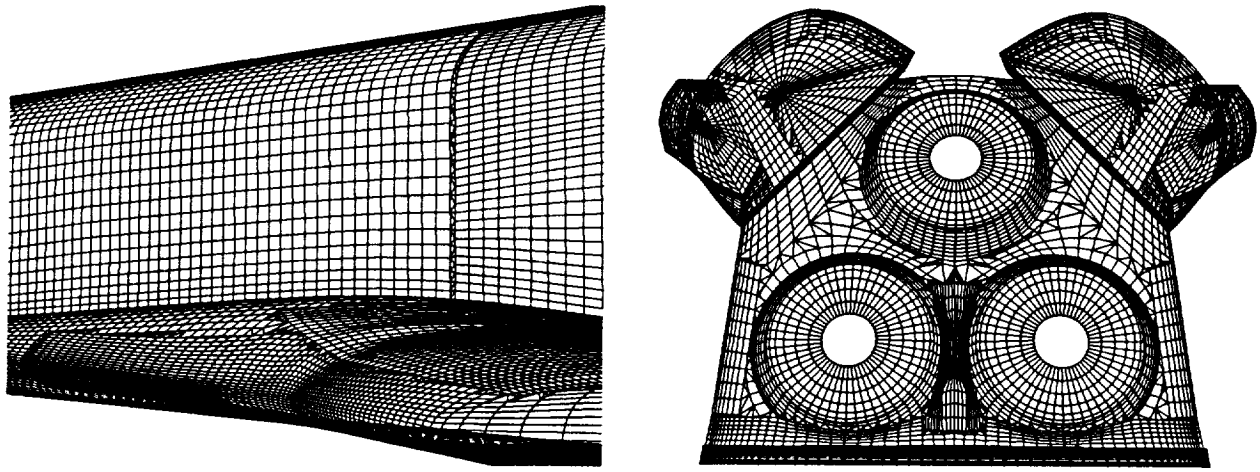
| Coefs | Databook | CFD-JSC | CFD-ZIP |
|---|---|---|---|
| $C_a$ | $0.1591 \pm 0.0319$ | 0.1339 | 0.1238 |
| $C_s$ | $0.0027 \pm 0.0119$ | 0.0026 | 0.0028 |
| $C_n$ | $-0.1438 \pm 0.0145$ | $-0.1472$ | $-0.1488$ |
| $C_{mr}$ | $-0.0005 \pm 0.0016$ | $-0.0001$ | $-0.0001$ |
| $C_{mp}$ | $0.0351 \pm 0.0097$ | 0.0273 | 0.0338 |
| $C_{my}$ | $0.0025 \pm 0.0067$ | 0.0016 | 0.0022 |

**Table 1.** Force and moment coefficients for the External Tank. The Databook and CFD-JSC values include the effects of the Orbiter/ET attach hardware but the CFD-ZIP value does not include such effects.



(a) (b)

**Figure 3.** Hybrid composite surface grids for the External Tank in the (a) forward attach hardware region, (b) base region looking from interior of tank to downstream.



(a) (b)

**Figure 4.** Hybrid composite surface grids for the Orbiter in the (a) wing leading edge region, (b) base region (nozzles not shown).

#### 4.1.2. *Orbiter*

The second comparison is performed on the Orbiter. Zipper grids for the fuselage/wing region and the base region are shown in Figures 4a and 4b, respectively. Comparisons with the Databook values and the JSC integration procedure are presented in Table 2. The results obtained using the current scheme are close to those obtained by the independent JSC integration program which shows that the procedure described in this paper is working fine. Discrepancies with the Databook values are probably caused by incomplete modelling of the physics by the flow solver, and difficulties in interpreting the uncertainty bounds in the Databook.

| Coefs | Databook | CFD-JSC | CFD-ZIP |
|---|---|---|---|
| $C_a$ | $0.0820 \pm 0.0161$ | 0.1101 | 0.1067 |
| $C_s$ | $0.0019 \pm 0.0100$ | 0.0004 | 0.0006 |
| $C_n$ | $0.0753 \pm 0.0267$ | 0.1023 | 0.0990 |
| $C_{mr}$ | $0.0015 \pm 0.0038$ | $-0.0007$ | $-0.0006$ |
| $C_{mp}$ | $-0.0341 \pm 0.0205$ | $-0.0537$ | $-0.0499$ |
| $C_{my}$ | $-0.0035 \pm 0.0075$ | 0.0005 | 0.0007 |

**Table 2.** Force and moment coefficients for the Orbiter.

#### 4.1.3. *Solid Rocket Boosters*

The next comparison is performed on the Left and Right Solid Rocket Boosters. Force and moment coefficients are given in Table 3a and 3b. Again, good agreement is obtained between the CFD-JSC results and the CFD-ZIP results.

| Coefs | Databook | CFD-JSC | CFD-ZIP |
|---|---|---|---|
| $C_a$ | $0.0349 \pm 0.0128$ | 0.0357 | 0.0348 |
| $C_s$ | $0.0265 \pm 0.0123$ | 0.0051 | 0.0007 |
| $C_n$ | $-0.0294 \pm 0.0103$ | $-0.0306$ | $-0.0308$ |
| $C_{mr}$ | $-0.0054 \pm 0.0018$ | 0.0060 | 0.0060 |
| $C_{mp}$ | $0.0034 \pm 0.0086$ | 0.0064 | 0.0066 |
| $C_{my}$ | $-0.0386 \pm 0.0094$ | 0.0131 | 0.0124 |

**Table 3a.** Force and moment coefficients for the Left Solid Rocket Booster.

| Coefs | Databook | CFD-JSC | CFD-ZIP |
|---|---|---|---|
| $C_a$ | $0.0348 \pm 0.0128$ | 0.0354 | 0.0343 |
| $C_s$ | $-0.0257 \pm 0.0123$ | $-0.0051$ | $-0.0009$ |
| $C_n$ | $-0.0278 \pm 0.0103$ | $-0.0289$ | $-0.0294$ |
| $C_{mr}$ | $0.0051 \pm 0.0018$ | $-0.0056$ | $-0.0058$ |
| $C_{mp}$ | $0.0005 \pm 0.0086$ | 0.0047 | 0.0052 |
| $C_{my}$ | $0.0390 \pm 0.0094$ | $-0.0132$ | $-0.0126$ |

**Table 3b.** Force and moment coefficients for the Right Solid Rocket Booster.

### 4.2. *Subsonic Transport Configuration*

This test case involves a subsonic transport configuration where the data was supplied by Dr. Lie-Mine Gea.[11] The geometry consists of a fuselage, wing, pylon, nacelle and cowl (see Figure 5). The overset grid system consists of grids for the above components together with collar grids for the pylon/wing junction, pylon/nacelle junction and the pylon/cowl junction. The numerical solution was obtained using the flow solver OVERFLOW.[10] Zipper grids generated for some of the regions are shown in Figure 6a,b. Force coefficients are computed by applying the procedure described in this paper on the numerical solution. Comparisons are made with data obtained from wind tunnel tests performed at the National Transonic Facility. For a lift coefficient of 0.61, the difference in drag coefficient between the wind tunnel data and the CFD result is 0.0003. However, an inviscid wall was assumed for the fuselage in the CFD run. Hence, the difference in drag coefficient is expected to be larger than that stated above.

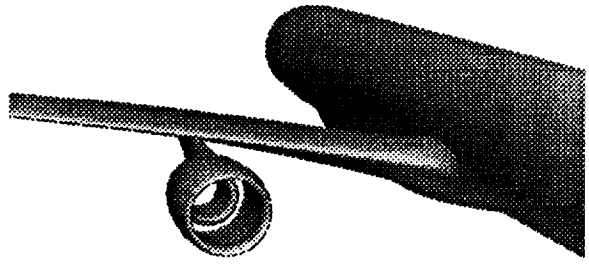
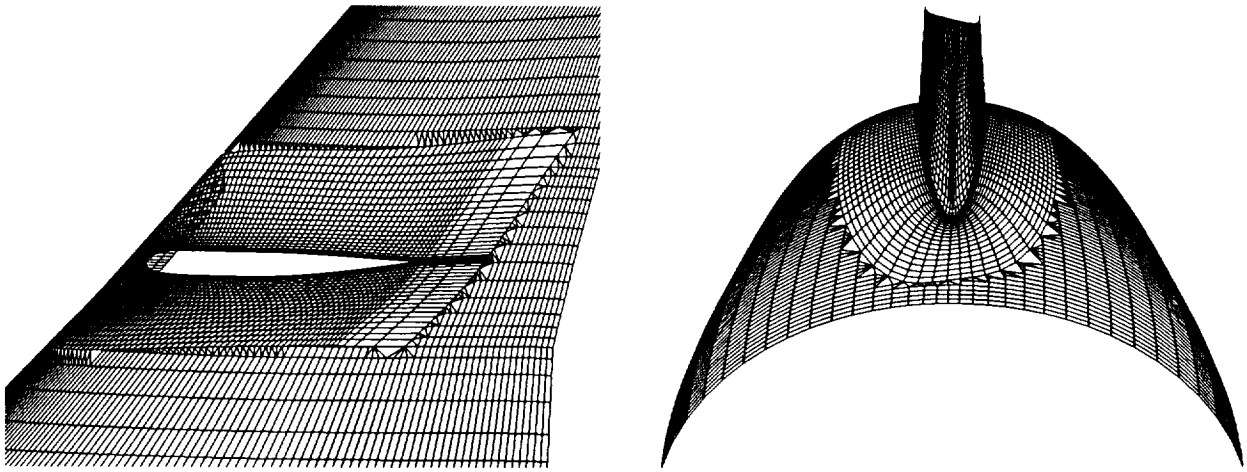


**Figure 5.** Surface geometry of subsonic transport configuration.

### 4.3. *Oblique All Wing Configuration*

## 5. Summary and Conclusions

A critical missing piece of technology for overset grid methods has been filled by the development of a scheme to account for the overlapped zones when computing force and moment coefficients on overset surface grids. The procedure is divided into two main steps and is performed by the two computer codes developed: MIXSUR to generate the hybrid composite surface grid and OVERINT to perform the integration. A simple quadrature rule is used to perform the integration on the hybrid composite surface grid which consists of non-overlapping quads and triangles that conform to the original surface. Very good comparison was obtained between data extracted from numerical solutions and wind tunnel tests for various complex configurations.



(a) (b)

**Figure 6.** (a) Hybrid composite surface grid in the lower wing and the pylon/wing collar grid region (top view). Part of the pylon/wing collar surface grid wraps over the leading edge of the wing as shown. (b) Hybrid composite surface grid in the upper outer nacelle and pylon/nacelle collar grid region.


## Acknowledgements

The authors would like to thank Mr. Ray Gomez of NASA Johnson Space Center and Mr. Jeff Slotnick of Lockheed Engineering for many detailed discussions on the Space Shuttle Launch Vehicle computations and for providing the corresponding data sets. Data for the subsonic transport configuration was kindly supplied by Dr. Lie-Mine Gea of McDonnell Douglas. This work is sponsored in part by the Space Shuttle Engineering Integration Office at NASA Johnson Space Center via NASA Ames Research Center, and in part by the Advanced Subsonic Technology Program at NASA Ames Research Center under consortium number NCC2-654.


## References


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

2. Gomez, R. J. and Ma, E. C., "Validation of a Large Scale Chimera Grid System for the Space Shuttle Launch Vehicle," AIAA Paper 94-1859, *Proceedings of the AIAA 12th Applied Aerodynamics Conference*, Colorado Springs, CO, June, 1994.

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

4. Martin, F. W., Labbe, S., Pearce, D. G. and Wey, T. C., "Space Shuttle Launch Vehicle Wind Tunnel and Flight Aerodynamic Environments," AIAA Paper 94-1861, June, 1994.

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

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

7. Dietz, W. E., "General Approach to Calculating Forces and Moments on Overset Grid Configurations," *Proceedings of the 2nd Overset Composite Grid and Solution Technology Symposium*, Northwest Florida Section of AIAA, Fort Walton Beach, Florida, October, 1994.

8. Kao, K-H., Liou, M-S., "Direct Replacement of Arbitrary Grid-Overlapping by Nonstructured Grid," AIAA Paper 95-0346, January, 1995.

9. "Operational Aerodynamic Databook," Rockwell Document STS85-0118 CHG. 4, Rockwell Space Systems Division, Downey, California, January, 1992.

10. Buning, P. G., Chan, W. M., Renze, K. J., Sondak, D., Chiu, I. T. and Slotnick, J. P., "OVERFLOW User's Manual, Version 1.6xx", NASA Ames Research Center, Moffett Field, CA, 1994.

11. Gea, L. M., Halsey, N. D., Intemann, G. A. and Buning, P. G., "Applications of the 3-D Navier-Stokes Code OVERFLOW for Analyzing Propulsion-Airframe Integration Related Issues on Subsonic Transports," ICAS Paper 94-3.7.4, 19th Congress of the International Council of the Aeronautical Sciences, September, 1994.

12. Chiu, I. T. and Meakin, R., "On Automating Domain Connectivity for Overset Grids," AIAA Paper 95-0854, AIAA 33rd Aerospace Sciences Meeting, January, 1995.

# Appendix C

*Abstract for presentation at the 2nd Overset Composite Grid and Solution Technology Symposium, Fort Walton Beach, Florida, October 25-28, 1994.*

# Recent Developments in Grid Generation and Force Integration Technology for Overset Grids

William M. Chan
MCAT Institute
NASA-Ames Research Center
Mail Stop T27B-2
Moffett Field, California 94035

## Abstract

Recent developments in algorithms and software tools for generating overset structured grids for complex configurations are described. Three areas of research will be addressed: surface grid generation, volume grid generation and force/moment integration.

## 1. Surface Grid Generation

The surface grid generation process for structured grids typically consists of the following two steps. The first step involves decomposing the surface geometry into topologically simpler domains and the second step involves generating grids on these domains that conform to the surface. Despite the availability of a wide selection of grid generation software packages, e.g. GRIDGEN, ICEMCFD, surface grid generation remains a very time consuming process for complex configurations. The main reasons for this are discussed below.

For the domain decomposition step, the time required to complete the task depends heavily on the experience of the user and the complexity of the geometry. As far as the author is aware, there is no software tool available that can automatically analyze the surface geometry and produce a satisfactory domain decomposition.

For the grid generation step, most of the surface grid generation tools available are based on the patched grid approach where the boundaries of adjacent blocks are required to abut each other. These tools utilize algebraic or elliptic methods where all four boundaries of the domain must be specified prior to surface grid generation. Since there are no general software packages available that are tailored for overset grids, overset grid users have been forced to employ patched grid tools to generate surface grids. However, the overset grid strategy suggests that many grids can be conveniently generated by specifying only one of the four boundaries - a task well suited for hyperbolic methods.

The development of the SURGRD code is an attempt to take a first step at constructing a surface grid generator tailored for overset grids. A basic version

was introduced by Steger[1] that generates a hyperbolic surface grid with a single panel network as the geometry definition. The hyperbolic scheme requires the specification of an initial curve on the geometry definition that is to be one of the boundaries of the desired surface grid. The grid is generated by marching away from the curve with the marching distance and step sizes prescribed by a stretching function. Recently, major new developments to the SURGRD code include the following.

(1) Complex configurations can now be treated by the admission of multiple panel networks as the geometry definition. Surface grids are generated and projected on to the networks. The networks are allowed to be abutting, overlapping or have small gaps in between.

(2) More sophisticated boundary treatments have been implemented to allow generation of a wider variety of grids. Options for the side boundaries include periodic, free floating, free floating along x, y or z constant planes, free floating along constant family lines on a panel network of the geometry definition. Limited control of the far field boundary shape is achieved via a variable marching distance for each point on the initial curve.

(3) A more robust projection algorithm has been implemented for the multiple panel network surface definition.

(4) An interpolation option is included which performs grid generation by marching along constant family lines of a panel network in the surface definition. This scheme is used in situations where a non-orthogonal marching direction from the initial curve is desired, e.g. the wing part of a wing/body collar grid.

Surface grids for the V-22 fuselage, wing and nacelle generated from a multiple panel network definition using SURGRD are shown in Figure 1. More details on the hyperbolic surface grid generation scheme and new developments on SURGRD are described in Ref. 2.

## 2. Volume Grid Generation

The volume grid generation step for overset grids is usually most efficiently achieved via hyperbolic methods. Recent developments for the hyperbolic volume grid generator HYPGEN (version 2.0) are described. The HYPGEN program is used to generate a 3-D volume grid over a user-supplied single-block surface grid by marching away from the initial surface with the step size given by a stretching function in the normal direction. This is accomplished by solving the 3-D hyperbolic grid generation equations (two orthogonality relations and one cell volume constraint).[3]

HYPGEN is a fully self-contained code and can be run in batch mode on any machine supporting Fortran 77. Users on Silicon Graphics (SGI) workstations have the option to use the graphical user interface HGUI for preparing and fine tuning the input parameters. Hooks to HYPGEN and PLOT3D are provided in HGUI for grid generation and visualization, respectively.

The main new features in version 2.0 of HYPGEN include

(1) For the stretching function in the marching direction, the initial spacing, final spacing and marching distance can be made to varying for each point on the surface grid. Further generality is achieved by having an option to read in a stretching function specified by the user.

(2) New boundary condition types are introduced to treat different grid topologies, e.g. collapsed edge for C or O grids at wing tips, constant interior x, y or z planes.

Figure 2a shows an airfoil C-mesh where grid lines in the wake are fanned out using the variable initial grid spacing option. This type of mesh is desirable over the conventional C-mesh if the wake outflow boundary overlaps another coarser mesh such as in the case of multi-element airfoils. The viscous spacing at the C-cut is avoided to allow better communication in the overlapped zone. Figure 2b shows slices of a volume grid for a wing C-mesh where the tip is closed using a collapsed edge. More details on HYPGEN and HGUI are described in the user's manual.[4]

## 3. Force and Moment Integration

An important area of technology currently lacking in overset grid methods is that there is no accurate way to integrate the pressure and viscous stresses on solid surfaces to obtain forces and moments information. This is because there is no simple scheme available to account for the overlapped zones on the surface grids. A scheme that can be used to solve this problem is described briefly below. More details are given in a paper submitted to the AIAA 12th CFD conference.[5]

Points in the overlapped zones between surface grids are first blanked out. For any pair of grids that overlap, grid points on the coarser of the two grids are blanked. Triangular cells are then generated to fill in the resulting gaps between neighboring grids without introducing new points. The result is a hybrid composite surface consisting of non-overlapping quadrilaterals and triangles where the quadrilaterals are just the unblanked cells in the original surface grids and the triangles are the newly generated triangular cells in the gaps. A function whose values are known at the grid points can then be integrated accurately on the hybrid composite surface.

Two software modules are developed to perform the above task. The first module named MIXSUR is used to generate the hybrid composite surface. The second module named OVERINT is used to integrate the pressure and viscous stresses to obtain force and moment coefficients on the composite surface. Input files required from the user include (1) a PLOT3D multiple volume grid file with iblanks indicating hole points generated from a domain connectivity program such as PEGASUS[6] or DCF3D[7], (2) a corresponding PLOT3D multiple grid solution file, and (3) an input parameters file. Files (1) and (2) are the input grid and output solution files, respectively, for the overset grid compressible Navier-Stokes flow solver OVERFLOW.[8] The input parameters file is used to specify the subsets from the different grids that make up the solid surface where forces and moments are required. Both the MIXSUR and OVERINT modules have been tested on a number of configurations

including the Space Shuttle Launch Vehicle (see Figure 3), a generic wing/body, a subsonic transport and a wing/pylons/nacelles/fins combination. Both modules are almost ready for beta release.

## References


1. Steger, J. L., "Notes on Surface Grid Generation using Hyperbolic Partial Differential Equations," Internal Report TM CFD/UCD 89-101, Department of Mechanical, Aeronautical and Materials Engineering, Univ. of Calif., Davis, 1989.

2. Chan, W. M. and Buning, P. G., "A Hyperbolic Surface Grid Generation Scheme and its Applications," AIAA Paper 94-2208, AIAA 25th Fluid Dynamics Conference, Colorado Springs, Colorado, 1994.

3. Chan, W. M. and Steger, J. L., "Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme," *Appl. Math. and Comput.* **51**, 181–205, 1992.

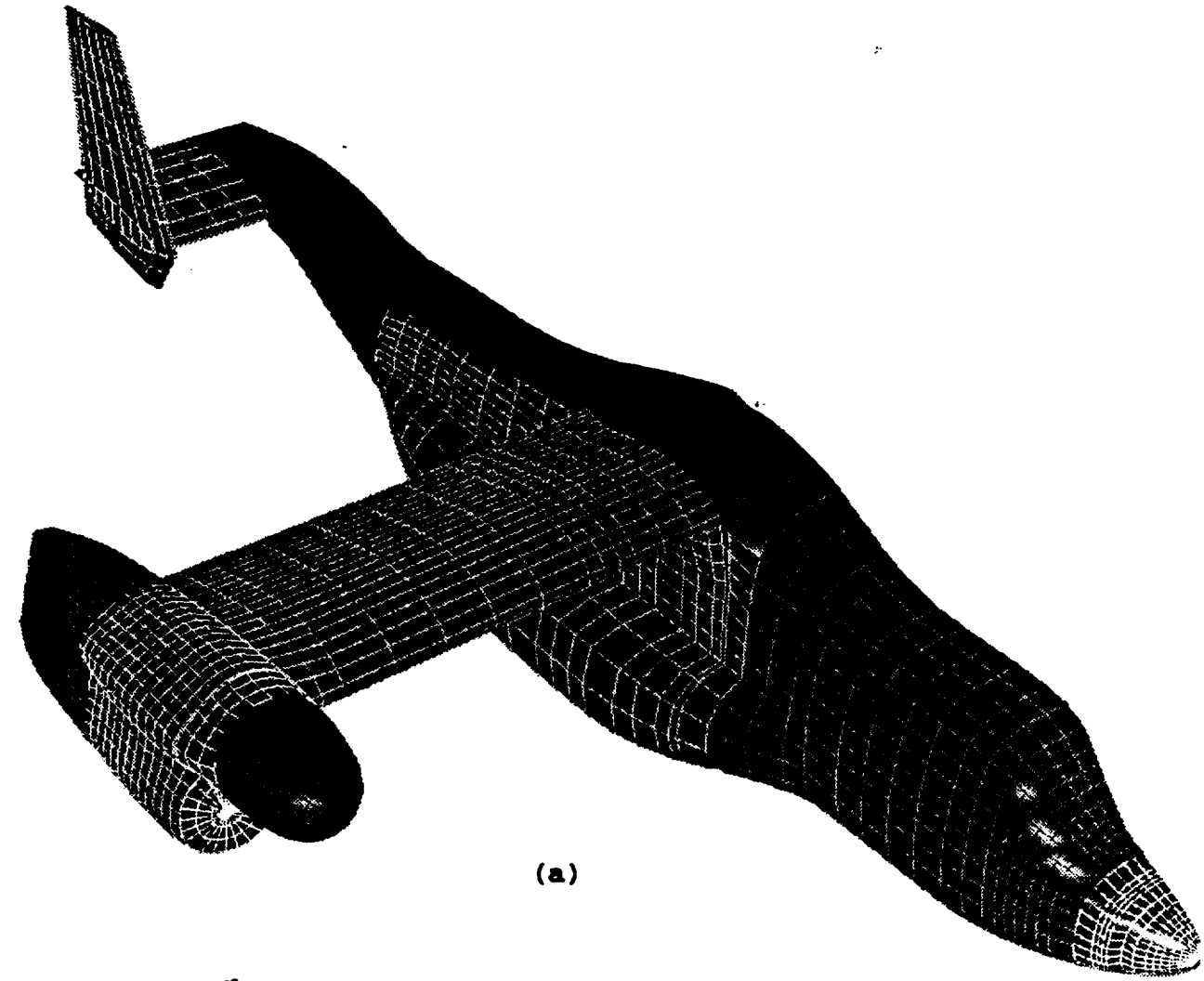
4. Chan, W. M., Chiu, I. T. and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, October, 1993.

5. Chan, W. M. and Buning, P. G., "Zipper Grids for Force and Moment Computation on Overset Grids," submitted to AIAA 12th Computational Fluid Dynamics Conference, San Diego, California, 1995.

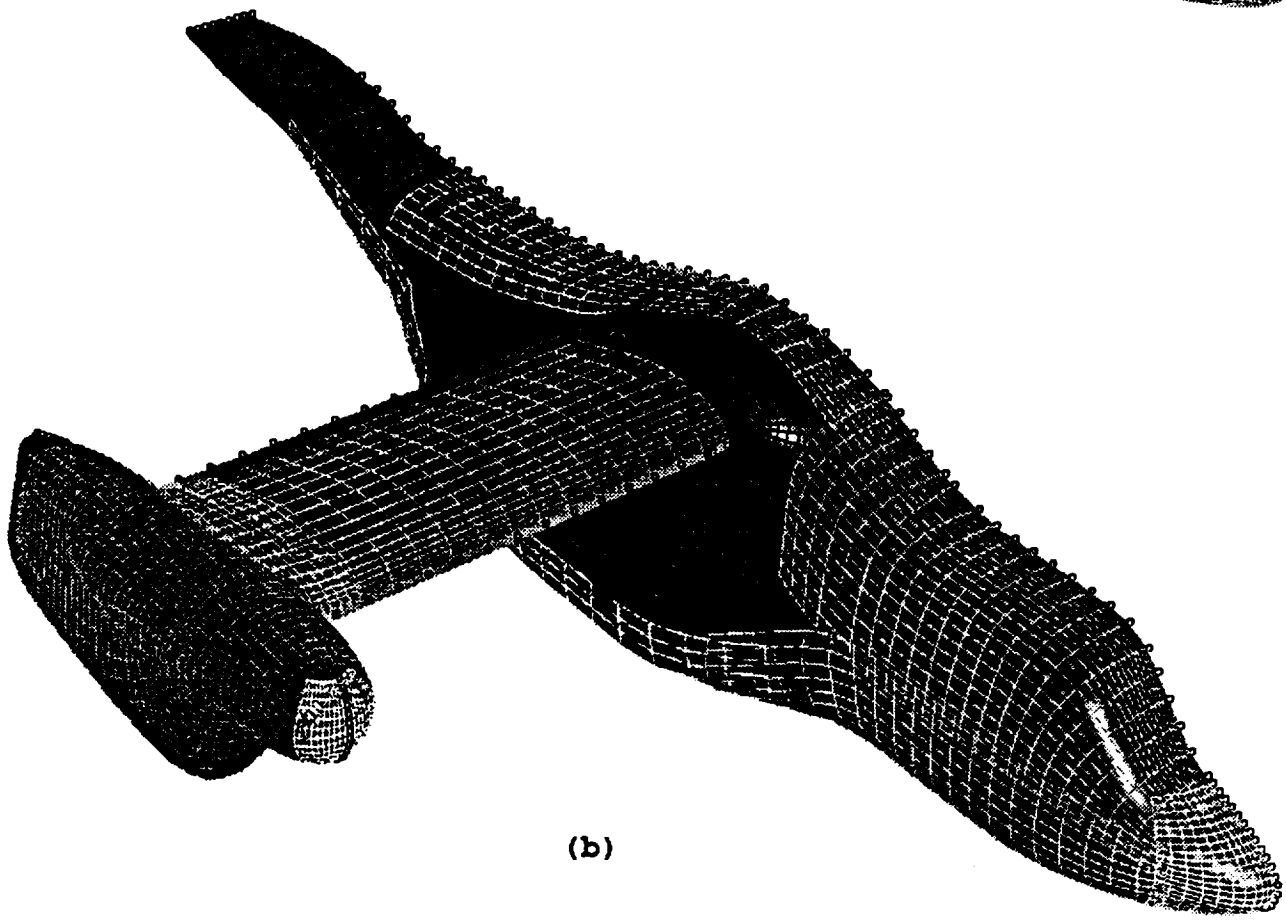
6. Suhs, N. E. and Tramel, R. W., "PEGSUS 4.0 User's Manual", AEDC-TR-91-8, June, 1991.

7. Meakin, R., "A New Method for Establishing Intergrid Communication Among Systems of Overset Grids," AIAA Paper 91-1586, *Proceedings of the AIAA 10th Computational Fluid Dynamics Conference*, Honolulu, Hawaii, 1991.

8. Buning, P. G., Chan, W. M., Renze, K. J., Sondak, D., Chiu, I. T. and Slotnick, J. P., "OVERFLOW User's Manual, Version 1.6xx", NASA Ames Research Center, Moffett Field, CA 94035, 1994.

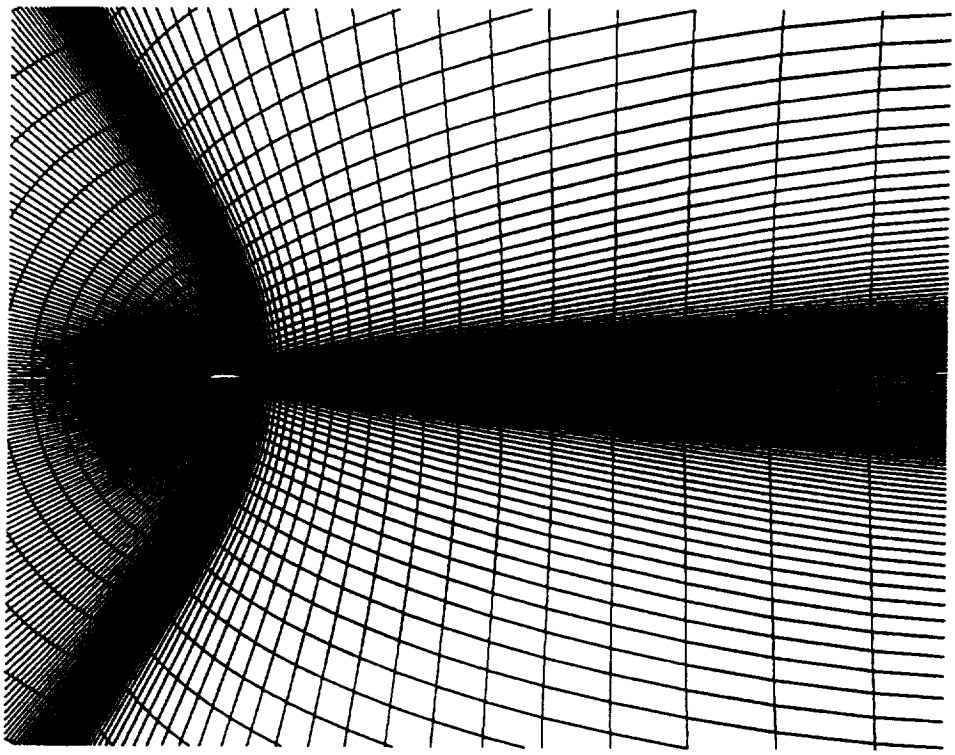


(a)

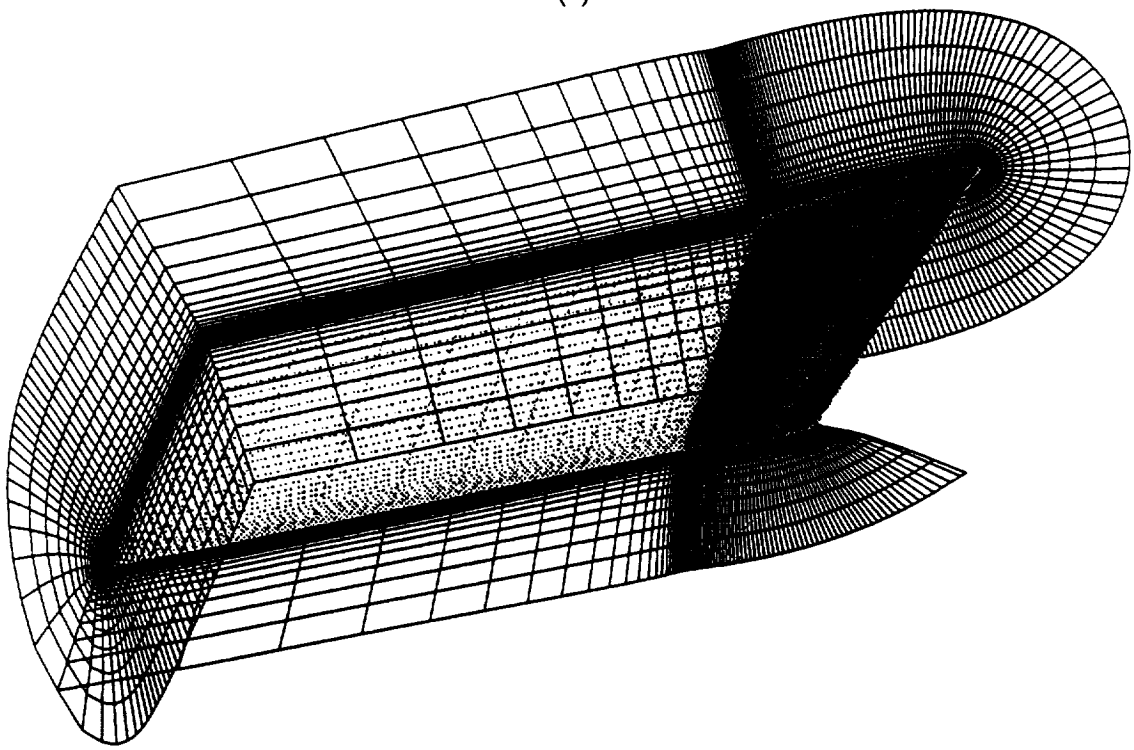


(b)

Figure 1 (a) Surface definition for the right half of the V-22 tiltrotor consisting of 22 panel networks. (b) Overset surface grids for the fuselage, wing and nacelle where points on the initial curves are indicated by circles.



(a)



(b)

**Figure 2.** (a) Airfoil C-mesh where grid lines in the wake region are fanned out using the variable initial grid spacing option. (b) Various computational planes of the volume grid for a wing (C-topology) with a collapsed edge boundary condition applied at the tip.

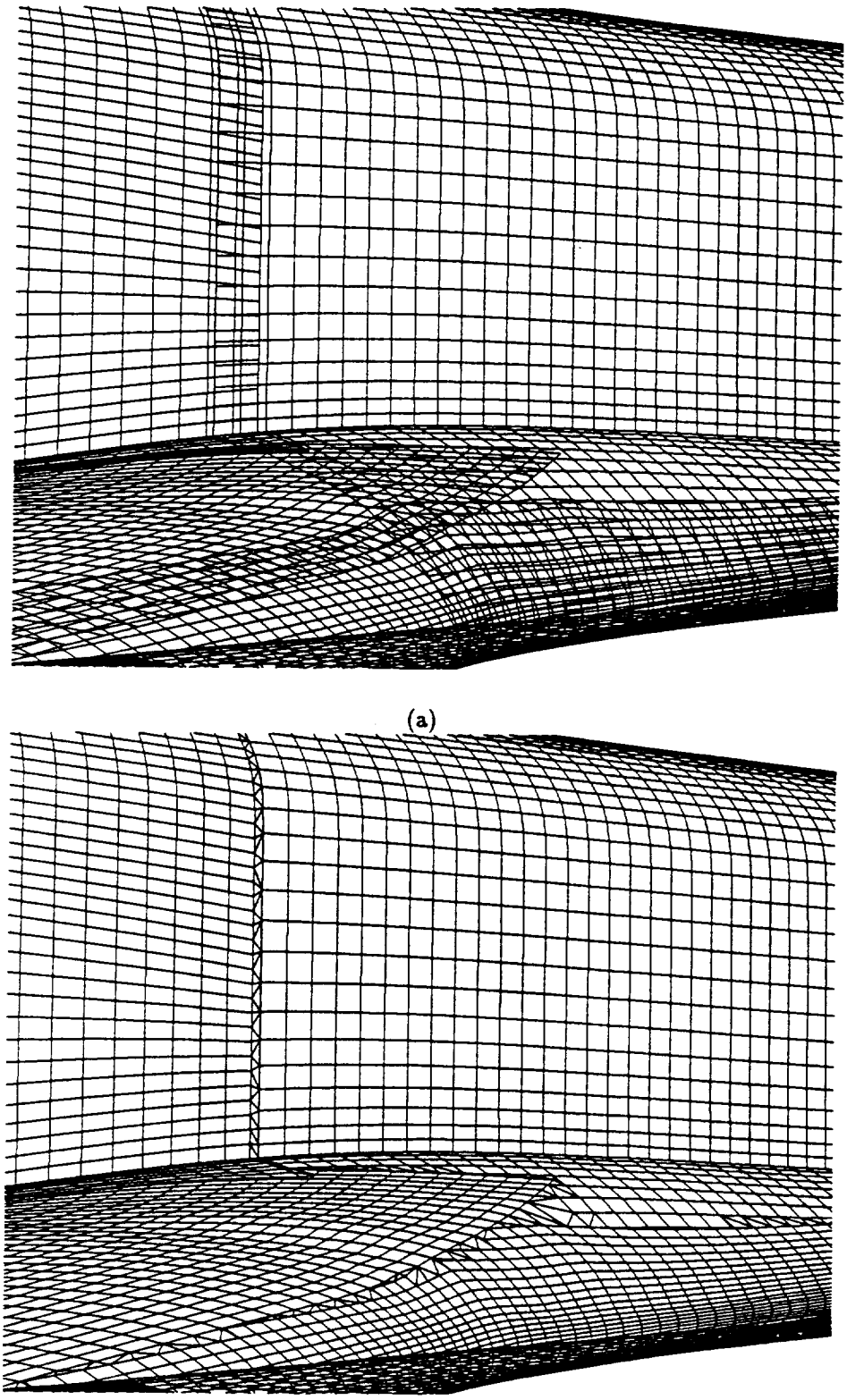


**Figure 3.** (a) Overlapping surface grids in the wing/fuselage region of the Space Shuttle Orbiter.
(b) Composite surface grid consisting of non-overlapping quads and triangles generated using MIXSUR.

# RECENT DEVELOPMENTS IN GRID GENERATION AND FORCE INTEGRATION TECHNOLOGY FOR OVERSET GRIDS

**William M. Chan**

***MCAT Institute***
***NASA Ames Research Center***
***Mail Stop T27B-2***
***Moffett Field, CA 94035***

2nd Overset Composite Grid and Solution Technology Symposium,
Fort Walton Beach, Florida, October 25-28, 1994

## SOME CRITICAL AREAS OF OVERSET GRID TECHNOLOGY

- Surface Domain Decomposition
- Surface Grid Generation
- Domain Connectivity
- Accurate Force and Moment Computation
- Grid Adaption
- Conservation Issues

# OUTLINE

- Overset Surface Grid Generation
- Hyperbolic Volume Grid Generation
- Force and Moment Computation
- Summary and Conclusions

# OVERSET SURFACE GRID GENERATION

***Motivation***

- Lack of surface grid generator tailored for overset grids
- Algebraic/elliptic methods not convenient for grids requiring one-sided specification and clustering control

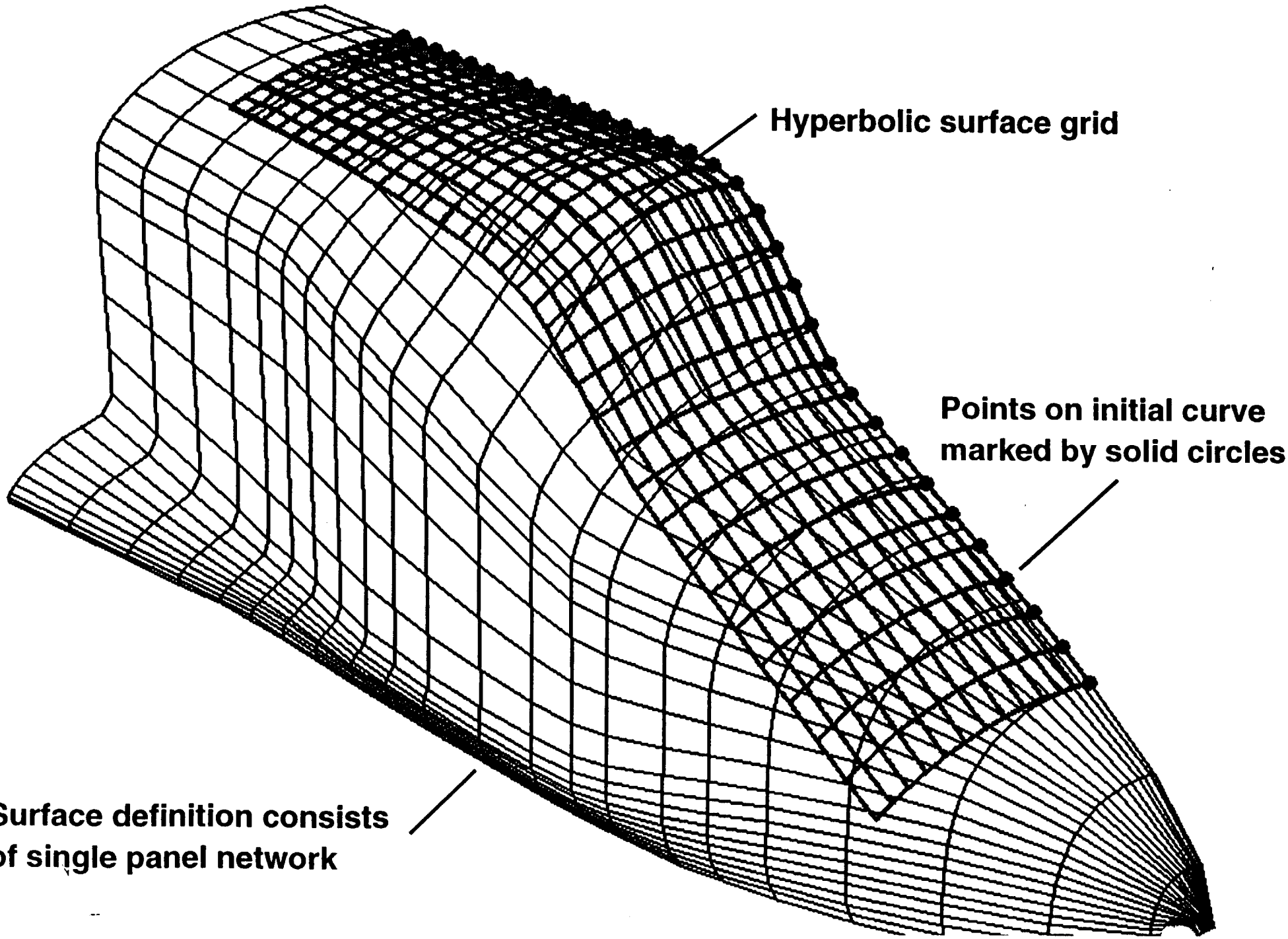
***Procedure***

(1) Define initial curves - control curves of surface geometry
(2) Redistribute points on initial curves
(3) Generate new layer of points by marching - hyperbolic or interpolation
(4) Project on to surface definition
(5) Repeat (3) and (4) until done

- Grid generation step and projection step are independent
- Publication

Chan, W. M. and Buning, P. G., "A Hyperbolic Surface Grid Generation Scheme and its Applications," AIAA Paper 94-2208, AIAA 25th Fluid Dynamics Conference, Colorado Springs, Colorado, June, 1994.

# BASIC HYPERBOLIC SCHEME BY STEGER



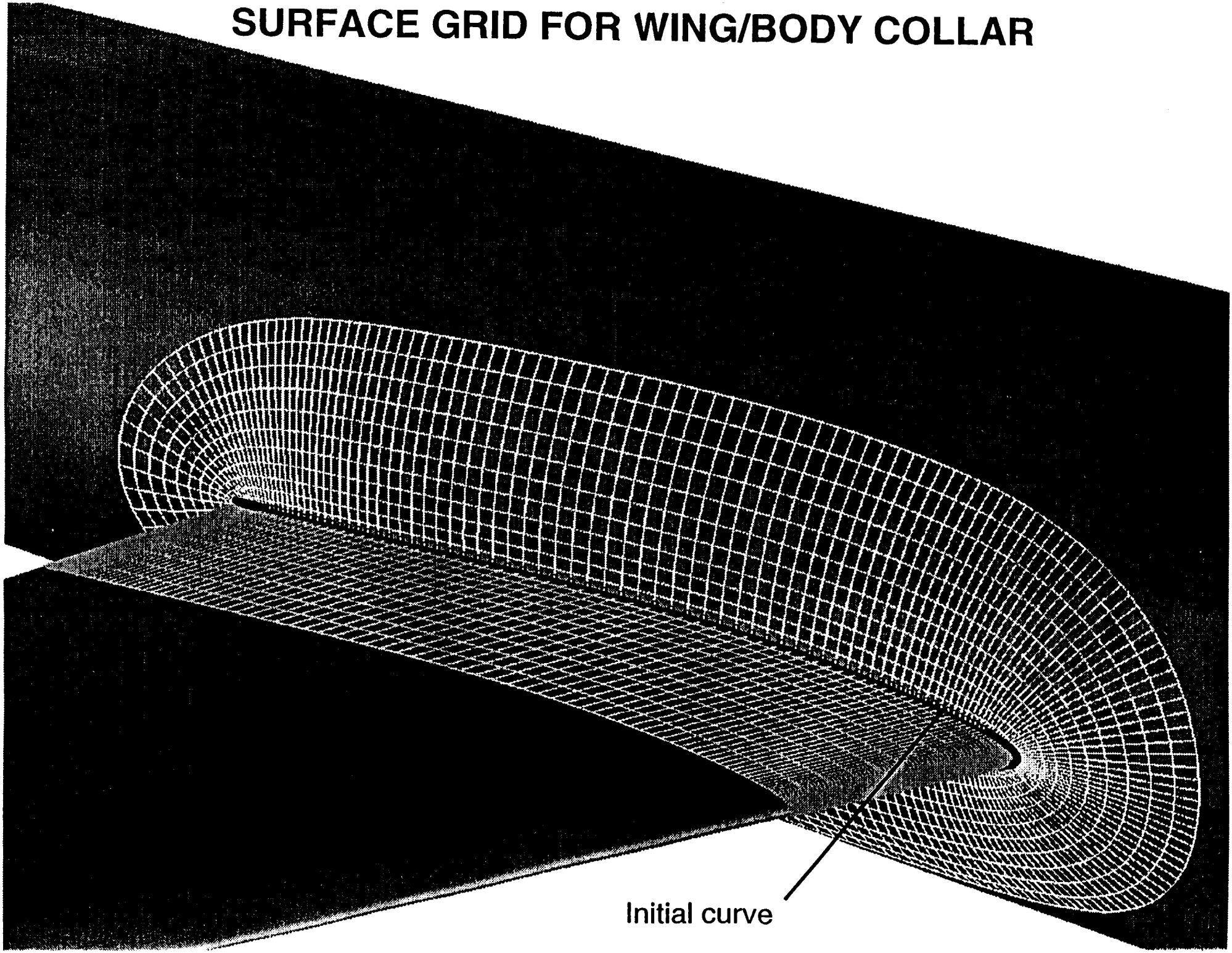
# OVERSET SURFACE GRID GENERATOR - SURGRD

- Geometry definition - collection of panel networks
  - Abutting, overlapping, intersecting, containing small gaps
  - Normals assumed consistent between networks
  - Auto-detect network connectivity
  - Bilinear surface assumed for projection
- Marching scheme - (1) Hyperbolic, (2) Interpolation
- Stretching control in marching direction
  - Hyperbolic tangent 1-D stretching function
  - Variable initial and end grid spacings
  - Variable marching distance - far field boundary control
- Side boundary control - boundary condition types
- Auto output of PLOT3D command files for visualization

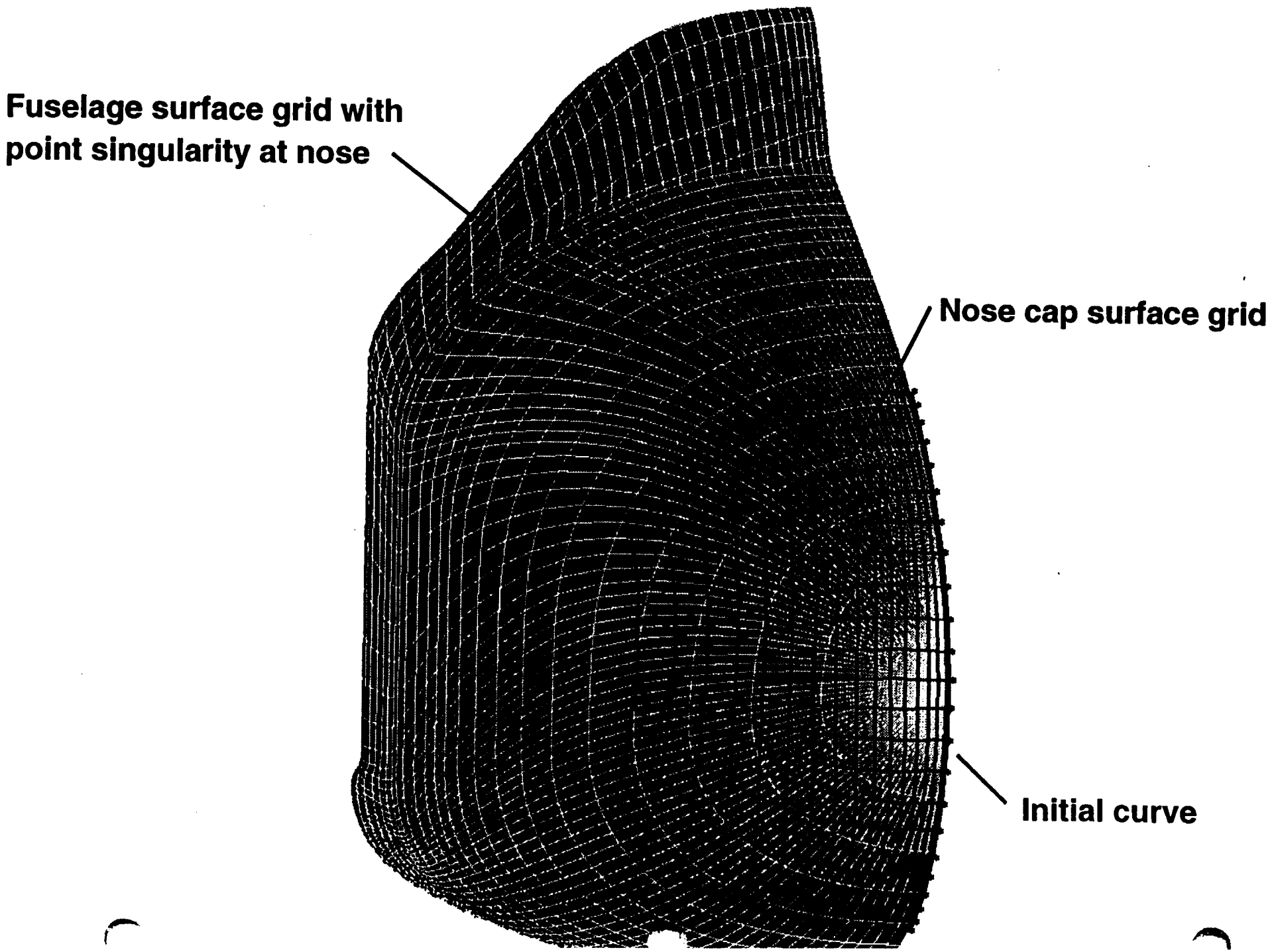
# APPLICATIONS

- Collar grids
  - Provide interpolation at component intersections
  - Examples: wing/body junction, wing/pylon junction
- Cap grids
  - Avoid singularities
  - Avoid highly skewed cells
  - Examples: fuselage nose, tips for wings and fins
- Quilt grids
  - Used on smooth parts of surface geometry
  - Need hyperbolic and algebraic capability
  - Examples: fuselage, wing

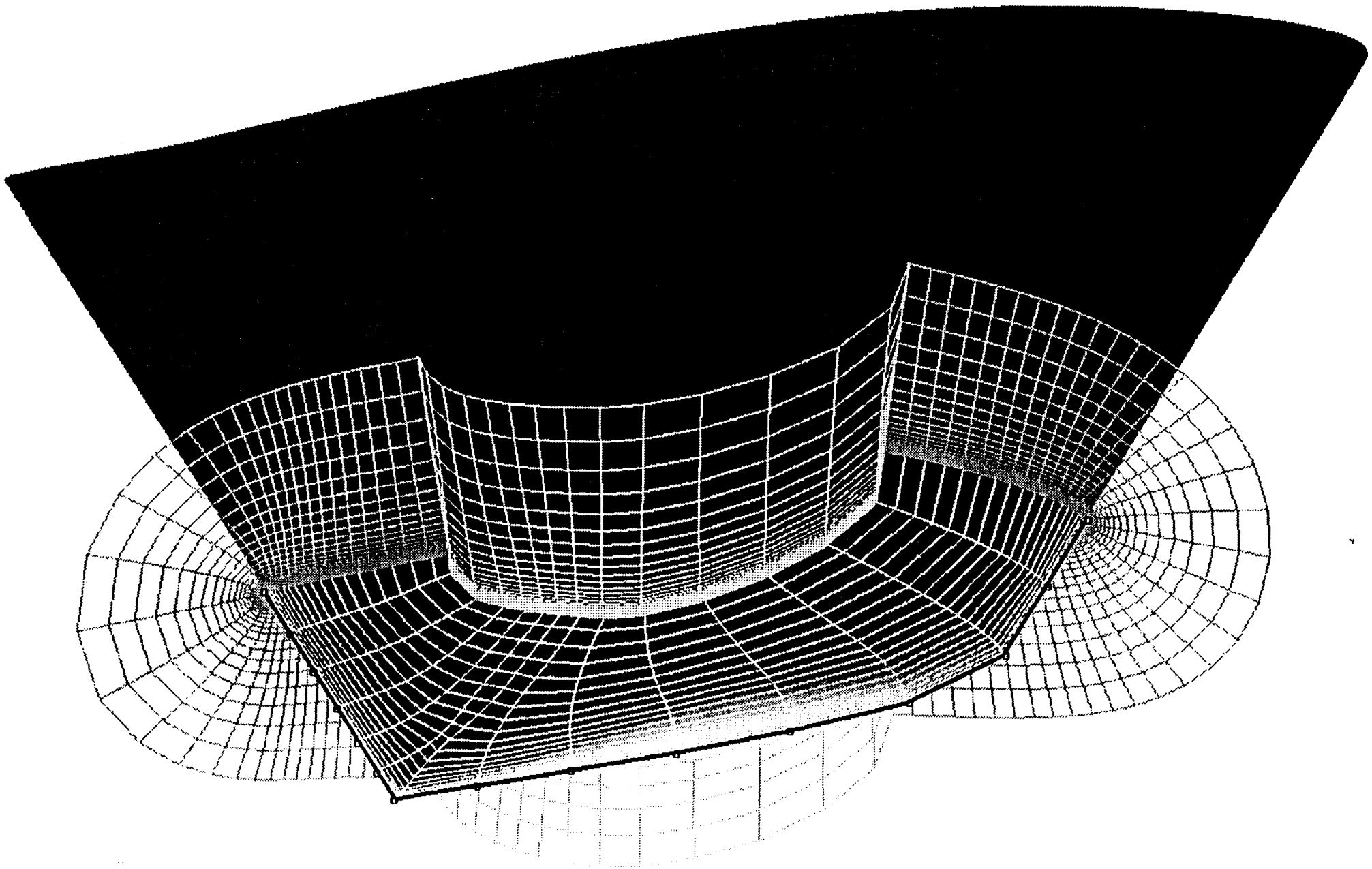
# SURFACE GRID FOR WING/BODY COLLAR



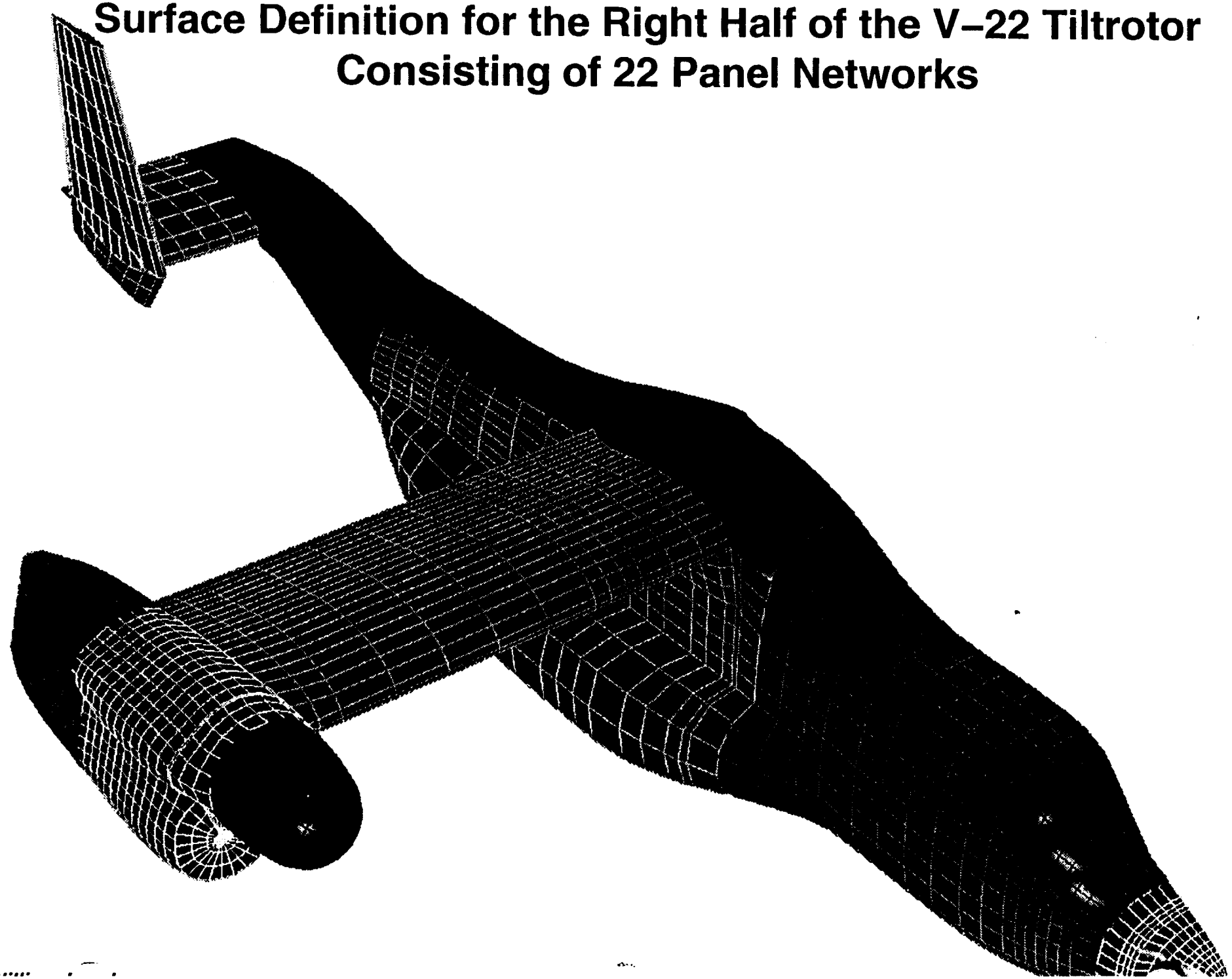
# SURFACE GRID FOR FUSELAGE NOSE CAP



# SURFACE GRID AND VARIOUS SLICES OF VOLUME GRID FOR WING TIP CAP

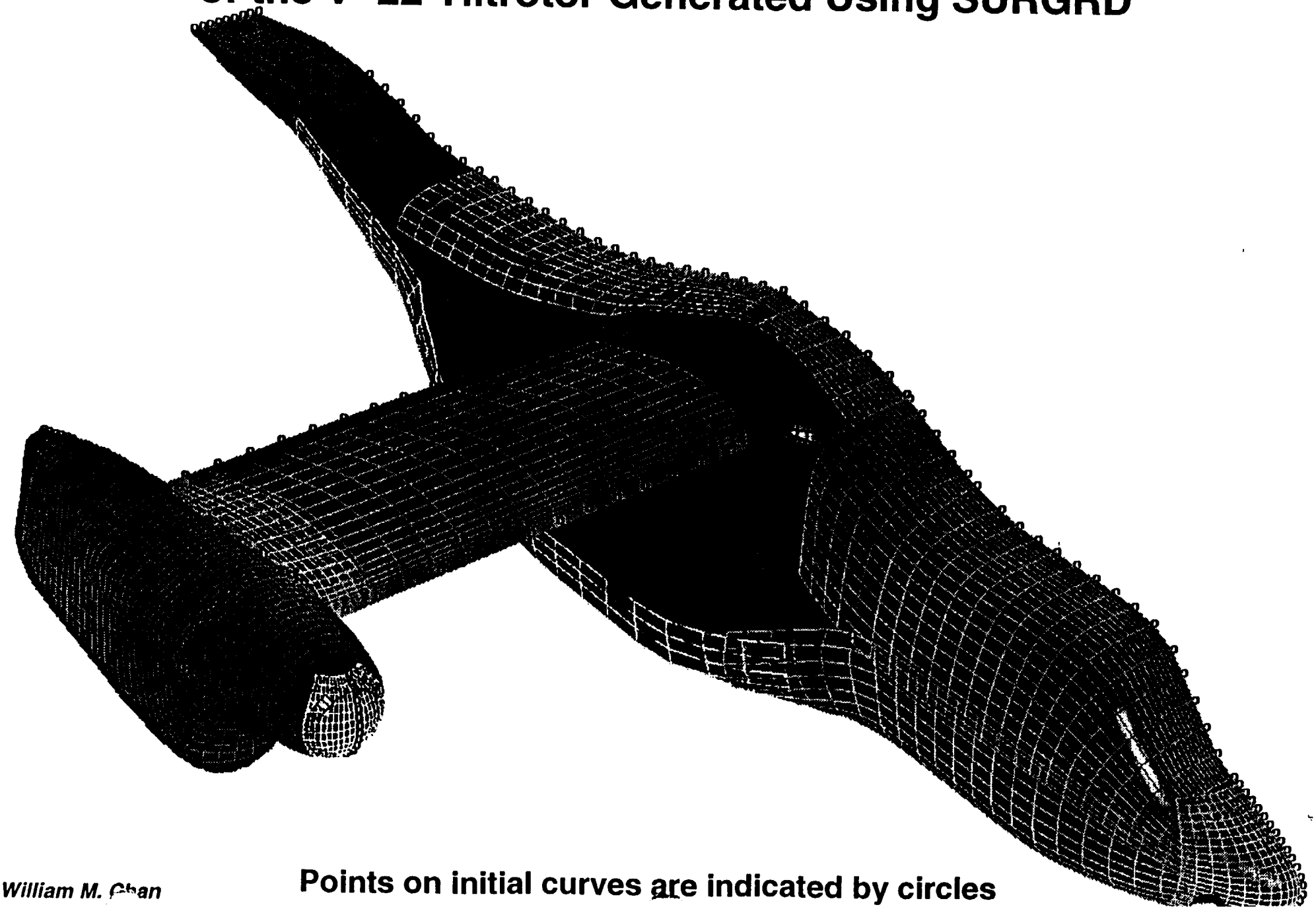


# Surface Definition for the Right Half of the V–22 Tiltrotor Consisting of 22 Panel Networks



# INITIAL CURVES DEFINED FOR V–22 TILTROTOR BEFORE SURFACE GRID GENERATION USING SURGRD

# Overset Surface Grids for the Fuselage, Wing and Nacelle of the V–22 Tiltrotor Generated Using SURGRD



Points on initial curves are indicated by circles

*William M. Chan*

# OVERSET VOLUME GRID GENERATION

- Hyperbolic methods - efficient and convenient
- Basic hyperbolic grid generation equations (Steger) - two orthogonality relations and one cell volume constraint
- Central difference in surface directions
- Implicit marching scheme in normal direction
- Spatially variable smoothing
- Implicit averaging at convex corners

# HYPGEN - BASIC FEATURES

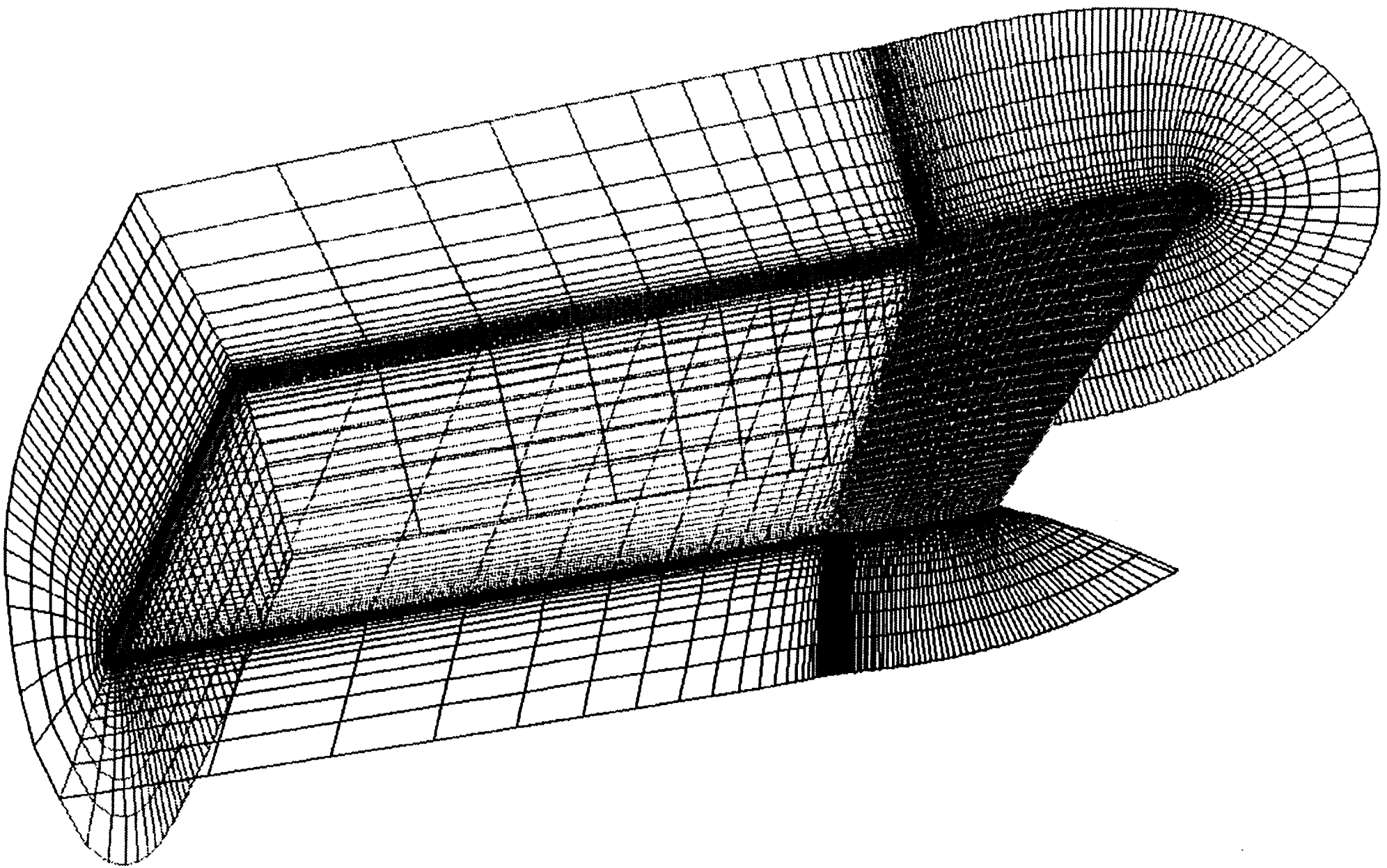
- Input
  - Surface grid
  - Stretching function in normal direction (step sizes, marching distances, number of points)
  - Smoothing parameters
- Output - volume grid generated by marching from surface
- Side boundaries controlled by boundary condition types
- Step sizes and marching distance controlled by stretching function
- Built-in grid quality checks
- Batch or graphical user interface (HGUI by I.T. Chiu)
- Speed - 230000 pts/s on C-90, 6000 pts/s on SGI R4000

# HYPGEN - VERSION 2.0

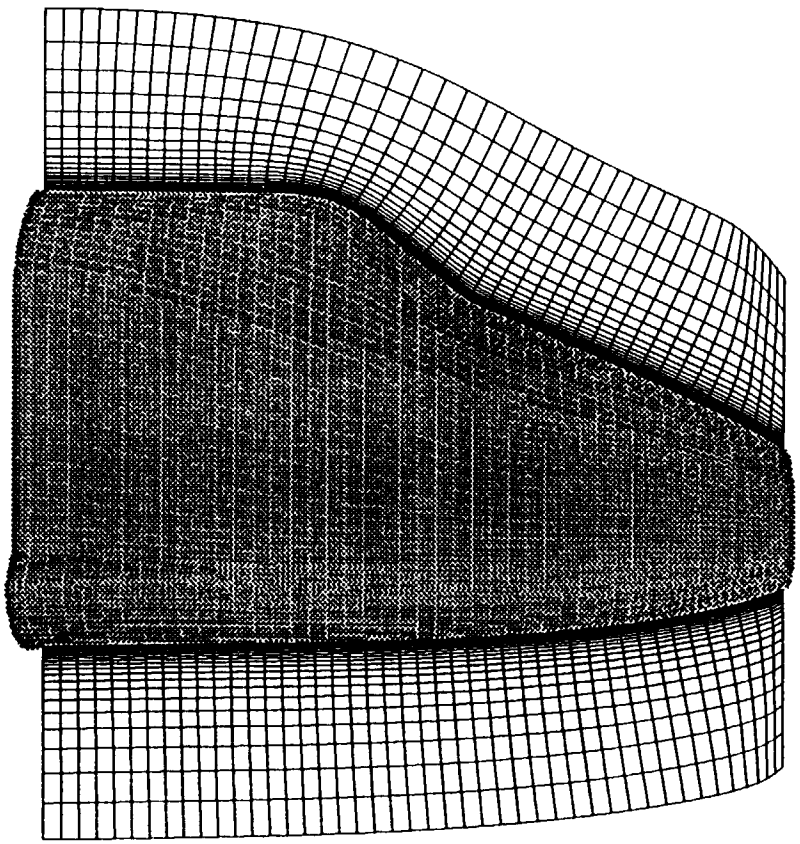
- New boundary condition types - collapsed edge, constant interior planes
- More robust constant plane boundary treatment
- Variable far field, initial and end spacing in normal direction
- Option to read in user-defined normal stretching function
- New set of examples
- Combined user's manual for HYPGEN/HGUI

  Chan, W. M., Chiu, I. T. and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, October, 1993.
- Optional volume grid generation module in NGP

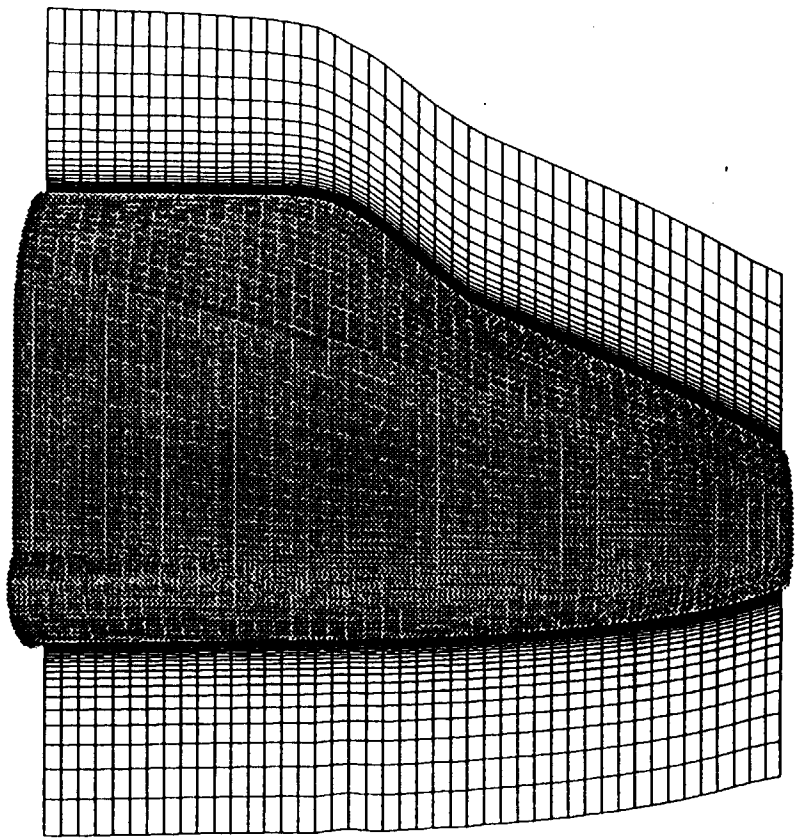
# Various Computational Planes of Volume Grid for Wing with Collapsed Edge at Wing Tip (C–mesh)



# CONSTANT PLANE BOUNDARY CONDITIONS

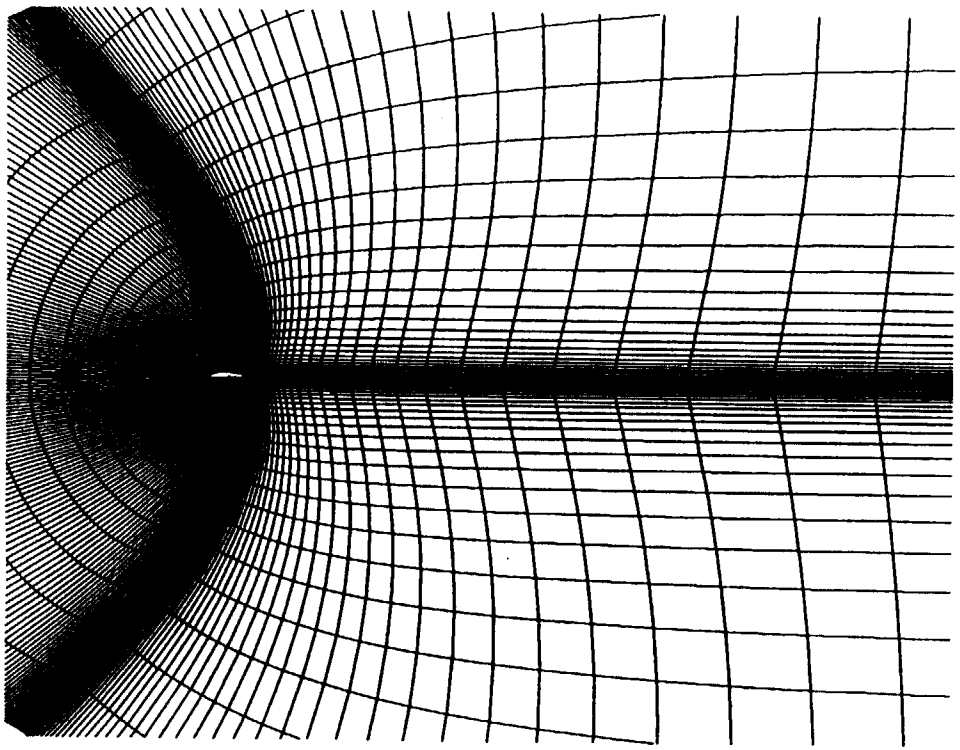


Constant boundary planes

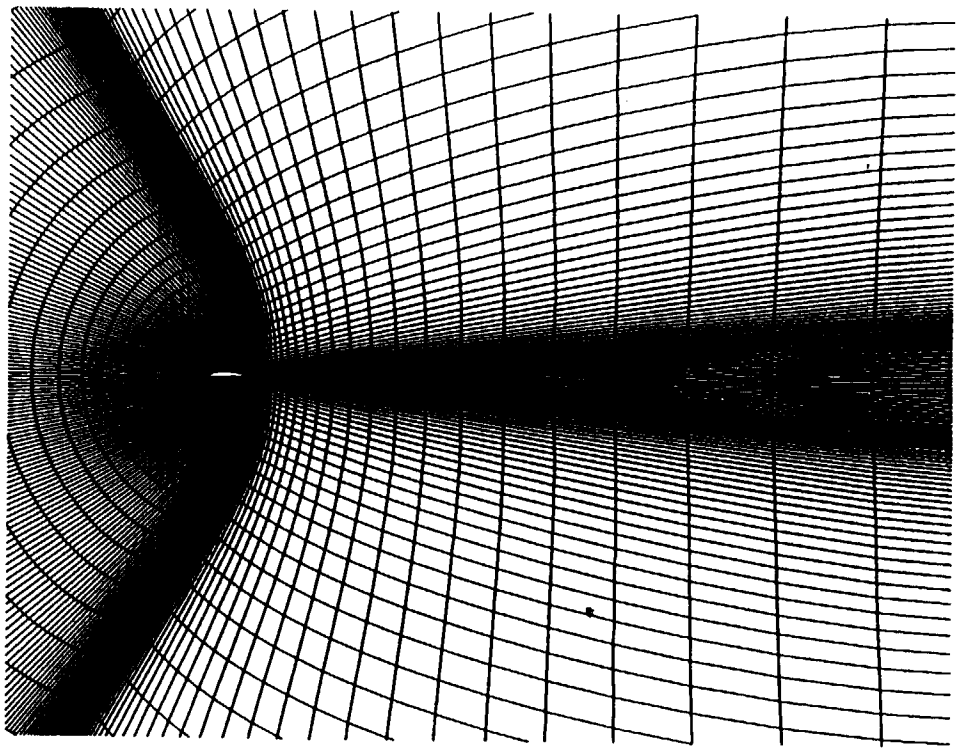


Constant planes everywhere

# AIRFOIL C-MESH



Constant wake spacing



Variable wake spacing

# FORCE/MOMENT COMPUTATION FOR OVERSET GRIDS

## *Motivation*

- Lack of accurate method to compute force and moment coefficients for overset surface grids
- Needed for aerodynamics performance computation - design, optimization, moving body problems

## *Procedure*

(1) Hybrid composite grid generation (MIXSUR)

- Performed once before flow solver call

(2) Pressure/viscous stress integration (OVERINT)

- Performed once after flow solver call or every N iterations

- Publication

Chan, W. M. and Buning, P. G., "Zipper Grids for Force and Moment Computation on Overset Grids," submitted to AIAA 12th Computational Fluid Dynamics Conference, San Diego, California, June, 1995.

## HYBRID COMPOSITE GRID GENERATION (MIXSUR)

- Surface integration domain consists of collection of subsets
- Each subset is a subset of a J, K or L constant slice of a volume grid

***Input***

- PLOT3D multiple volume grid file with iblanks
- Input parameters file to specify surface subsets

***Procedure***

(1) Blank points in overlapped regions
-> gaps

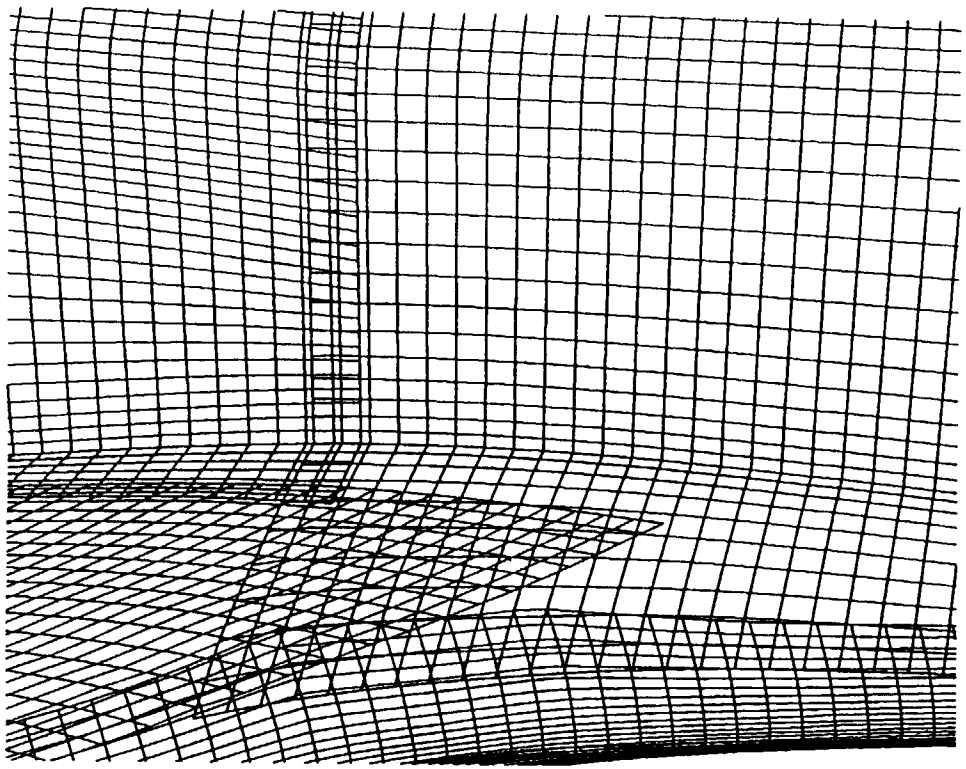
(2) Identify and order points on gap boundaries
-> gap boundary strings

(3) Match and connect points between gap boundary strings
-> hybrid composite surface grid

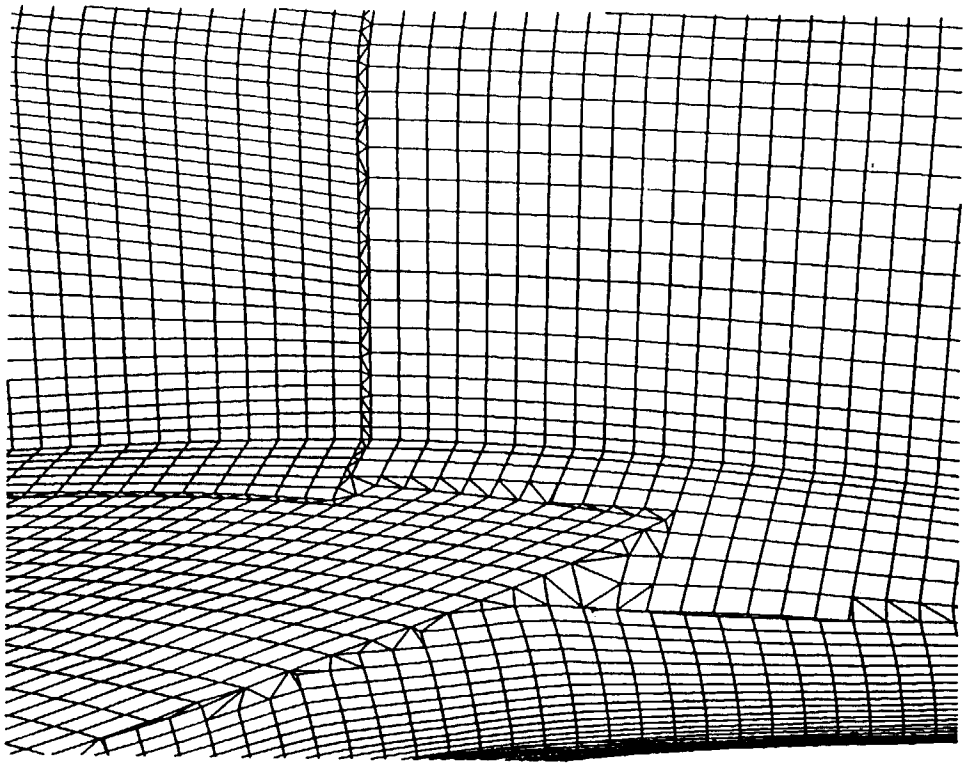
***Output***

- Surface with non-overlapping quads and triangles

# HYBRID COMPOSITE SURFACE GRID GENERATION



Overset surface grids



Hybrid composite surface grid

## PRESSURE AND VISCOUS STRESS INTEGRATION (OVERINT)

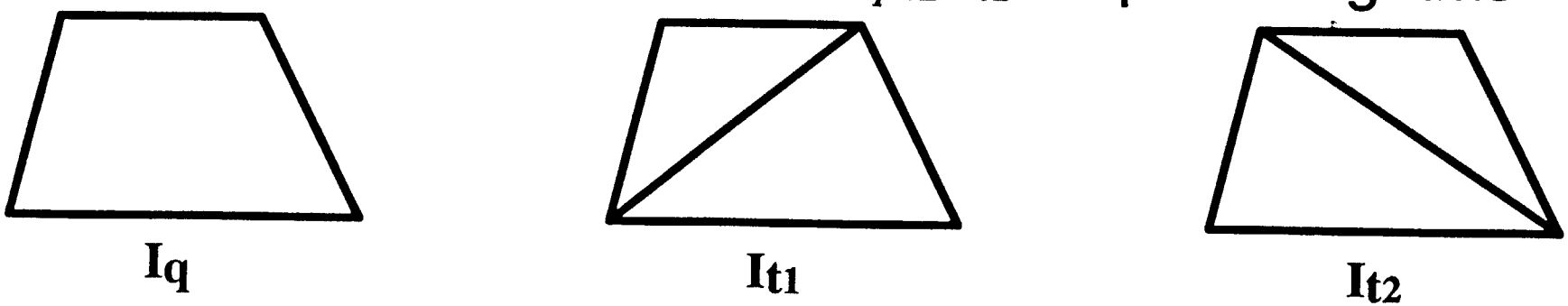
- Performed on hybrid composite surface grid

***Input***

- Files produced by MIXSUR
- PLOT3D multiple grid solution file
- Same input parameters file as MIXSUR

***Quadrature***

- Assume piece-wise constant function inside quad/triangle
- Constant = average of function at vertices
- Three ways to do quads, $I_{t1} \leq I_q \leq I_{t2}$ for parallelograms



$I_q$ $I_{t1}$ $I_{t2}$

***Output***

- Force/moment coefficients, integrated areas

# FORCE/MOMENT COEFFICIENTS FOR ORBITER

| Coefficients | Databook | CFD-JSC | CFD-ZIP |
|---|---|---|---|
| Axial Force | $0.0820 \pm 0.0161$ | 0.1101 | 0.1067 |
| Side Force | $0.0019 \pm 0.0100$ | 0.0004 | 0.0006 |
| Normal Force | $0.0753 \pm 0.0267$ | 0.1023 | 0.0990 |
| Rolling Moment | $0.0015 \pm 0.0038$ | $-0.0007$ | $-0.0006$ |
| Pitching Moment | $-0.0341 \pm 0.0205$ | $-0.0537$ | $-0.0499$ |
| Yawing Moment | $-0.0035 \pm 0.0075$ | 0.0005 | 0.0007 |

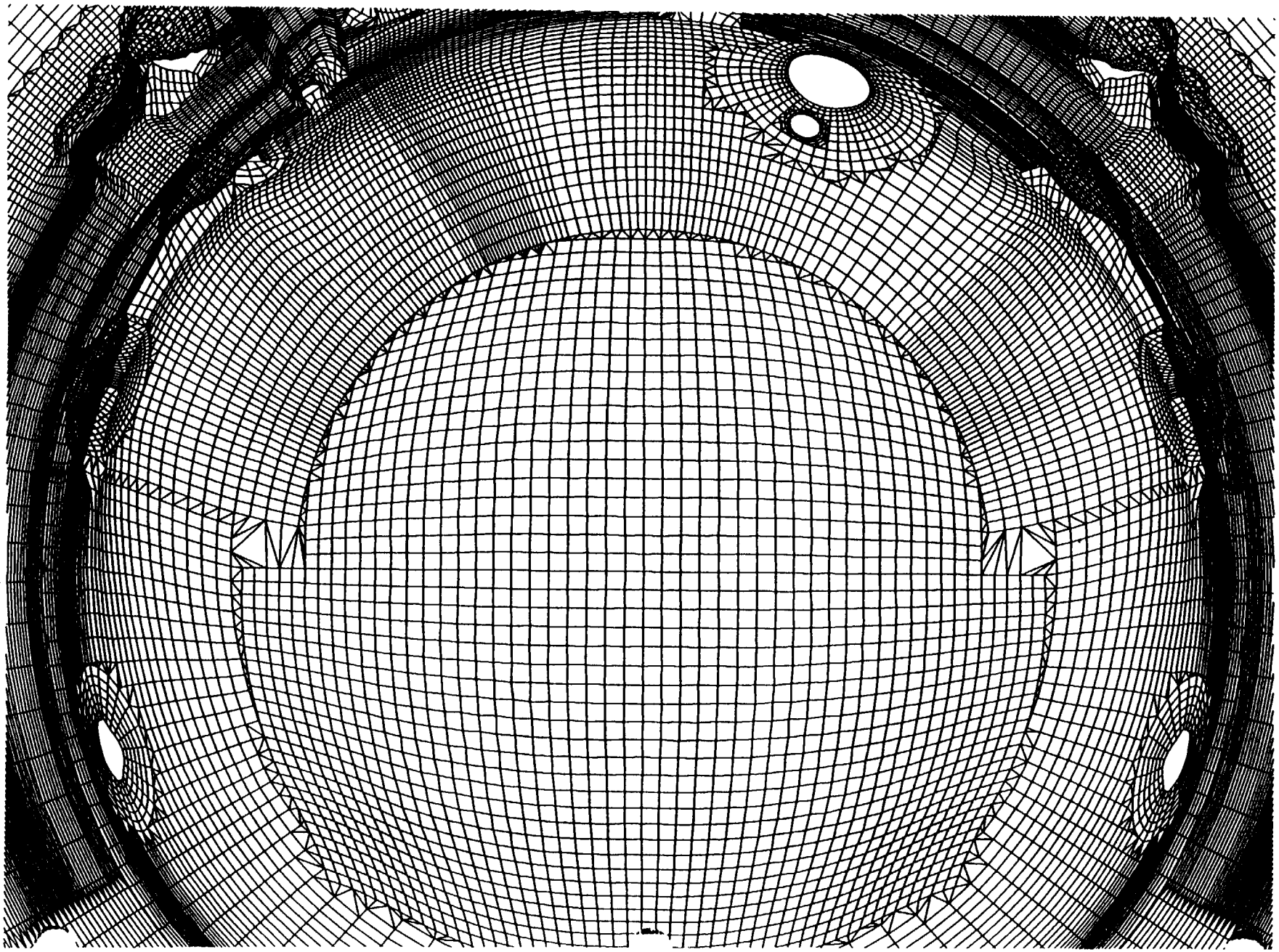
Databook = Rockwell wind tunnel data adjusted for flight Reynolds number ($\pm$ gives uncertainties)

CFD-JSC = independent integration procedure at NASA JSC

CFD-ZIP = current zipper grid procedure

# HYBRID COMPOSITE SURFACE GRID IN EXTERNAL TANK BASE REGION

View from inside looking downstream



# SUMMARY AND CONCLUSIONS

## *SURGRD*

- A first step towards an overset surface grid generator
- Enhancements lead to treatment of more complex surfaces
- Reduction in surface grid generation time, potential for partial automation

## *HYPGEN*

- An easy to use hyperbolic volume grid generator
- Version 2.0 - more grid types, more robustness, optional module in NGP

## *MIXSUR, OVERINT*

- Missing gap in overset grid technology filled by development of accurate force/moment computation scheme

  (1) Hybrid composite grid generation (MIXSUR)

  (2) Pressure and viscous stress integration (OVERINT)

# STATUS AND PLANS

## *SURGRD*

- Status - beta release
- Plans - ability to use NURBS as geometry definition

## *HYPGEN*

- Status - production
- Plans - more robustness and automation

## *MIXSUR, OVERINT*

- Status - almost beta release
- Plans - scheme to remove bad triangles, more accurate blanking scheme, incorporation into OVERFLOW