NASA Technical Memorandum 108986

SINGLE BLOCK THREE-DIMENSIONAL VOLUME GRIDS ABOUT COMPLEX AERODYNAMIC VEHICLES

Stephen J. Alter

K. James Weilmuenster

November 1993

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

Abstract

This paper presents an alternate approach for the generation of volumetric grids for supersonic and hypersonic flows about complex configurations. The method uses parametric two-dimensional block face grid definition within the frame work of GRID-GEN2D. The incorporation of face decomposition reduces complex surfaces to simple shapes. These simple shapes are recombined to obtain the final face definition. The advantages of this method include the reduction of overall grid generation time through the use of vectorized computer code, the elimination of the need to generate matching block faces, and the implementation of simplified boundary conditions. A simple axisymmetric grid is used to illustrate this method. In addition, volume grids for two complex configurations, the Langley Lifting Body (HL-20) and the Space Shuttle Orbiter are shown.

PAGE [redacted] INTENTIONALLY [redacted]

Contents

Introduction

A large portion of the engineering (i.e. wall clock) time required to do a CFD computation about complex aerodynamic configurations is consumed in the grid generation process. Currently, three-dimensional volume grid generators available in the public domain offer a variety of options for grid generation for highly complex aerodynamic configurations. To simplify the grid generation process, 3D grid solvers usually rely on multiple block volume decompositions. Each block then requires simple algebraic/elliptic grid solutions. Although this approach has been used extensively, it has some inherent disadvantages.

The most significant disadvantage to multiple block volume decompositions is the necessity to generate matching block face boundaries which are required to generate the initial grid using three-dimensional trans-finite interpolation[1] (3DTFI). The requirement is that all six faces of a grid block volume must be previously defined by two-dimensional parametric surface grids. Typically, for inviscid Computational Fluid Dynamic (CFD) solutions, 3DTFI produces an adequate volume grid. However, for viscid CFD solutions orthogonality is usually required at the wall of a configuration. The 3DTFI method of volume grid initialization does not guarantee orthogonal grid lines where appropriate (i.e., at the wall, symmetry planes, etc.). To obtain the required orthogonality, an elliptic solution of Poisson's equations is used. Popular algorithms in Poisson solvers, utilized for slope continuity across matching block interfaces, significantly alter the original defining interface surface (Fig. 1). The popular EAGLE[2] and GRIDGEN3D[3] codes, compute a solution between two blocks to determine the "correct" location of a matching boundary. Utilizing this solution results in a different location and distributions of grid points on a matching block interface (Fig.

1). Thus, for multiple block decompositions, the interfaces of matching blocks have to be initially defined, but the time used to generate them is lost when they are altered by the elliptic solver.

The inability to effectively use vectorized code is a second disadvantage. The short vector length and coded conditionals associated with multiple block decompositions degrade the performance of vector processors. Also, the memory management techniques required to locate multiple blocks in array space and locate points required to compute the location of the matching interface after each solver iteration, further limits the effectiveness of vector processors.

A third disadvantage to using multiple block topologies is the possibility of non-linear and cyclic oscillations in grid point movement residuals leading to elliptic solver divergence. The unstable nature of these oscillations can be the result of several categories. These categories will be further identified later, as well as the techniques for alleviating the induced oscillations.

The fourth disadvantage to using multiple blocks is the necessity to use a surface grid generator to obtain the matching block interface. As the complexity of the configuration increases so does the matching block interface. For the Shuttle Orbiter, these surfaces can be located between the configuration's wall and the outer shock boundary as well as vertical tail and fuselage intersections. Simple surfaces obtained from planar cuts or TFI usually do not produce adequately smooth interface surfaces. Rather, complex surface generators have to be used to obtain these matching block interface surfaces which adds more time to the grid generation process.

In order to generate three-dimensional volume grids for highly complex configurations,

it is advantageous to use a blocking topology that requires the least number of blocks. This paper presents an alternate approach to developing and constructing single block three-dimensional volume grids about complex aerodynamic vehicles explicitly for CFD. The technique is difficult compared to multiple block decompositions, but results in a significant savings of nearly 50 percent in the engineering time required to develop a grid for a given configuration. A simple sphere-cone-cylinder-flare configuration is used to illustrate the advantages for single block volume grid refinement. Volume grids for the Langley Lifting Body (HL-20) and full Space Shuttle Orbiter are presented as illustrations of grids for complex configurations created using this approach.

Comparison of Blocking Strategies

Single Block

Typically, for a single block volume grid topology, a C type grid is used for both wings and fuselage. The only strategy for generating the volume grid is the development of parametrically two-dimensional grids on the six block boundaries (faces). A simple ellipsoidal (sphere-cone-cylinder-flare) configuration is utilized in this paper to illustrate the technique.

The overall strategy for generating the six individual faces of the single block can be found in the flowchart (Fig. 2) which is used because it embodies the grid design philosophies of the GRIDBLOCK/GRIDGEN2D[3] codes. Basically, the configuration is first broken into the six individual faces that represent the single block flowfield structure. Each face is broken into individual domains, as required, to represent some, if not all of a face, i.e. subface decomposition. Then each individual subface is generated utilizing a variety of algebraic and elliptic methods available within the GRIDGEN codes. This task is completed in an

interactive workstation environment where the user can visually and accurately place and cluster grid points.

After the subfaces and subsequent complete faces are generated, the boundary surfaces are transferred to a supercomputer where a three-dimensional elliptic solver is used to generate a volume grid. The elliptic solvers usually employ 3DTFI for grid initialization and a Poisson solver for elliptic smoothing and control.

Generally, the most complicated surface to generate is the configuration's surface. For the example case, illustrated in figure 3, the surface has been broken into four individual subfaces. The first subface represents the nose of the configuration. The second represents the conical portion, the third a cylindrical part, and the fourth is represented by the flare. Construction is carried out through a series of algebraic and elliptic solutions to obtain a slope continuous set of grid lines over the entire surface. All other faces are constructed similarly, using the subface decomposition to simplify the surfaces.

After the faces of the single block structure are defined by combining the various surface grids, the Three-Dimensional Multi-block Advanced Grid Generation System[4] (3DMAGGS) was used to generate the volume grid. The solver required 680 seconds of Cray-II CPU (Central Processing Unit) time to complete 400 iterations and a final grid point movement indicative of a converged solution. Figure 4a shows the RMS (root-mean-square) residual grid point movement for each iteration and figure 4b shows the RMS residual of the corrections to the orthogonality source terms for each iteration. The solution is considered to be converged when the average correction of the orthogonality source terms for each grid point is more than four orders of magnitude less than the largest orthogonality source term for the entire volume grid. The overall grid generation time, including the definition portion, and the

elapsed time from running the elliptic solver on a Cray-II, was 2 hours.

Multiple Block

The multiple block decomposition was constructed according to figure 5. The faces of each sub-block were generated in the same way that the faces of the single block grid were generated. However, the grid points on either side of the matching block interfaces were obtained in a different manner. These grid points had to be generated such that slope continuous grid lines would result across the matching block interface. Within the GRIDGEN framework, slope continuous lines are obtained by first copying the set of subfaces into a single face. This single face is solved as a complete entity resulting in slope continuity at the matching edges. Then the proper subfaces of the single face are inserted back into their original positions. The steps required to enhance slope continuity across matching boundaries and the need to match face boundaries increases the time required to define a multiple block grid over that required for a single block grid. For the example case, the multiple block grid required $3\frac{1}{2}$ hours to define the block boundaries while the single block only required $1\frac{1}{2}$ hours.

The multiple block example case required 820 seconds of Cray-II time to obtain 400 iterations of the 3D elliptic solver. Figure 6a shows the grid point movements for each block. An inspection of the maximum point movement for the multiple block volume grid—where the maximum movement for any of the 3 blocks is compared to the single block maximum point movements—shows that the multiple block movement residuals are slightly higher at the end of 400 iterations. Comparing the maximum grid-point movements between the 2 different block volume grids, figure 4a for the single block continuously converges after 15

iterations, but the multiple block, shown in figure 6a, diverges slightly during the first 40 iterations then starts to converge. Figure 6b shows the RMS of the orthogonality source term corrections. Again, as indicated for the single block grid, the corrections are about four orders of magnitude smaller than the largest orthogonality source term, which is an indication of a converged solution. Although the grid is converged, the solver required 21 percent more CPU time to generate a multiple block volume grid. Figure 7 shows a set of cross-sectional planes from the multiblock grid construction. The engineering time required to generate the three block volume grid was $4\frac{1}{2}$ hours, 225 percent over the time required to construct a single block grid, and both exhibited the same grid quality as shown in figure 8.

A break down of the time required for elements of the grid generation process for each gridding strategy is shown in figure 9. Although the single block method requires less time for all elements, the major difference in the two techniques is the time required for matching block interface construction for the multiple block approach.

A grid solution is considered to be converged when the movement of grid points in the volume per iteration is small for relaxation rates of 0.9 to 1.1. During the solution process for both the single block and multiple block grids, elliptic solver instability, as indicated by non-linear oscillations in the grid-point movement residuals, became a problem. This instability could be the result of several factors, which may be any or all of the following:

[1] Large relaxation rates for grid-point movement.

[2] Regions of strong surface gradients (i.e. high curvature and discontinuities).

[3] Regions of tight cell spacing near a surface where orthogonality is being enforced.

[4] Using forward or backward differences at a matching boundary.

Typically, this instability is characterized by an unbounded grid-point movement from iteration to iteration. The new computed location for the grid point can be controlled directly by limiting the amount of change i.e. relaxation rate, in the PSOR algorithm.

The other elements that may contribute to elliptic solver instability are more complex, making it difficult to counteract their affects. For multiple block grids, the computation at matching block interfaces poses the most difficulty because the interfaces usually occur in regions of surface geometry changes. These surface geometry changes typically occur at abrupt changes in surface curvature. When derivatives are computed at matching boundaries where the surface changes occur, forward or backward differences have to be used because the interface is the limit of a parametric direction. The forward and backward differences usually magnify the surface gradients as well as tight grid spacings which tends to produce orthogonality source-terms that change rapidly from point to point in either of the two para-

metric directions along a surface. Large fluctuations in the orthogonality source-terms used in Poisson's equations tend to cause grid-point movements to fluctuate. To test this phenomenon, a planar-surface grid was extracted from the single block solution for the multiple block grid, which extended from the intersection of the cylinder and flare to the outer flow domain. This defined the matching boundary interface for the multiple block grid. The multiple block volume grid was elliptically solved with identical cell sizes required for orthogonality, and the interface was allowed to float. Divergent cyclic oscillations in grid-point movement per iteration occurred at the wall of the configuration near the interfacing surface. Only the derivatives had changed between the two blocking topologies, suggesting that instability occurred when the central differencing, which smears out wall-surface gradients, was switched to the one-sided differencing.

Hence, matching block interfaces in regions of large surface gradients, and tight cell spacings tend to cause elliptic solver instability. The instability caused by these surface and grid spacing effects can be alleviated by averaging the orthogonality source-terms in the parametric direction and increasing the cell spacings at the wall. Both of these techniques were used for the multiple block volume grids generated for this paper.

Elliptic solver instability may also be caused by conflicting forcing functions near the matching boundary. In 3DMAGGS, slope continuity across matching boundaries is obtained using weak orthogonality controls on the matching interface. These orthogonality boundary conditions add another source-term to Poisson's equation for points near the matching boundary, and may conflict with the orthogonality specification at the configuration's surface. To reduce the effects of adding the source-term to obtain slope continuity, the decay rate of the source-terms into the volume interior was increased, which maintained near matching

boundary slope continuity and reduced the conflict on the interior of the volume grid.

A comparison of the single block elliptic solver residuals to those for the multiple block grid, shows the source-term correction RMS for the multiple block grid does not drop as quickly as that for the single block grid. The difference in convergence rate may be a result of transients created by grid initialization in the solution of the multiple block grid at the beginning of the solution. An inspection of the source-term correction residuals for the multiple block grid indicates transients characterized by cyclic oscillations in the beginning of the solution. But as the solution progresses the oscillations disappear and the multiple block grid continues to converge as evidenced by the decreasing values of the grid-point movement RMS residual. Although the single block grid has four spikes in the source-term correction convergence history, these transients are minor compared to the source-term oscillations encountered in the multiple block grid solution. Damping of these transients in the single block grid was quick, but the multiple block grid was affected by the oscillations over a larger number of iterations. These transients appear to be the result of tight spacing of grid points near the surface which are required for orthogonality control and were alleviated by increasing the cell sizes.

Application to Complex Geometries

Shuttle Orbiter

In this section, the application of single block grid construction for the Shuttle Orbiter will be demonstrated. In addition, a multiple block Orbiter grid has been constructed for comparison to the single block grid.

Single Block Orbiter:

The single block construction of the Shuttle Orbiter includes all of the geometry, except the engine nozzles. This complex configuration was constructed with the main surface and wing of the Shuttle as one face and with the vertical tail embedded in the leeside symmetry plane. The subface decomposition of the surface as shown in figure 10, was designed to isolate various portions of the Orbiter's surface based on surface grid requirements. These include the canopy region, the leading edge of the wing, the Orbital Maneuvering System (OMS) pods, the vertical tail, the leeside intersection between the wing and the fuselage, and the impact location of the bow shock onto the wing at high angles of attack. The leeside symmetry plane was decomposed into four subfaces to account for the vertical tail (subface 2) and its intersection with the fuselage (Fig. 11a). The exhaust plane of the flowfield was broken into three subfaces due to the highly complex nature of the defining edges (Fig. 11b). The only other difficulty encountered was lining up the grid point locations on the outer boundary (shock wave) surface such that the grid lines from the Orbiter's surface would be nearly straight and orthogonal to the configuration wall. The construction of the blocking and grid definitions of each face accounted for 30 hours.

After the faces were constructed, elliptic grid generation proceeded smoothly. The only

grid-point movement oscillations then tended to cause elliptic solver divergence occurred at the intersection of the vertical tail and the fuselage. To alleviate the oscillations the elliptic solver's orthogonality source-terms were averaged for discontinuous surface regions. The volume grid solution required 14.2 CRAY-II CPU hours utilizing 6800 iterations. Figure 12 shows the residual grid point movements for each iteration and figure 13 shows the orthogonality source term residuals for each iteration. Evaluating the elliptic solver residuals in both figures 12a and 12b, it is evident that the grid point movement jumps at certain intervals, indicated by the circled regions. These regions are restarts of the 3DMAGGS code, with different relaxation parameters. Due to previous calculations, experience has indicated that the solver is unstable with large initial relaxation parameters (on the order of 0.1 to 1.0) for this computation. The solutions were initially started with relaxation parameters of 0.01, then increased to its maximum of 0.4. This under relaxation rate was the closest to neutral stability, allowing the quickest convergence without induced instability.

A representative set of cross-sectional grid planes are shown in figure 13, with an expanded view of the three cross-sections in figures 14, 15, and 16, respectively. The inset in figure 16, shows the vertical tail has bi-directional viscous spacing on it. The corner where the fuselage, vertical tail and leeside symmetry plane meet was the area that limited the relaxation rate for the entire volumetric solution. The solver was limited due to the viscous spacing required to model both the boundary layer at the fuselage surface and the vertical tail surface. The entire grid generation process consumed 80 hours of engineering time, as shown in figure 17.

Multiple Block Orbiter:

The multiple block volume grid construction was more tedious. In addition to defining the 6 faces that comprised the single block grid, ten other matching faces had to be generated (Fig. 18). The grid generation time for the all boundaries was 120 hours.

The 3D elliptic generation of the multiple block volume grid was more involved, especially in the control of slope continuity across matching boundaries and cyclic oscillations in grid-point movement near the matching block boundaries. The 3DMAGGS code did not fair well with the matching boundaries. Slope continuity across each boundary was obtained by enforcing orthogonality with a rapid decay rate of the forcing functions into the volume grid. Due to the non-linear grid-point movements caused by forcing the solver to obtain slope continuity across matching interfaces, the solver forcing functions had to be averaged in the wing tip regions and cell sizes at the matching interfaces near the wall had to be increased. Similar to the single block topology, the forcing functions on the vertical tail intersections with the symmetry plane and fuselage were also averaged to obtain convergence. The time required to get the same number of iterations from the elliptic solver as used for the single block grid was 25.0 CRAY-II CPU hours. Figure 19a shows the grid point movement residuals, figure 19b shows the orthogonality source term residuals for each iteration and figures 20, 21, 22, and 23 show the same cross-sectional planar grids as shown for the single block grid.

When the elliptic solver performances are compared, the multiple block volume grid required fewer iterations to converge than did the single block grid. This may be due to the matching boundary interfaces giving more definition for initial volume grid construction.

Although extensive forcing function modifications were required to obtain a usable solution, the volume grid is comparable to the single block topology as shown in figure 24. The total grid generation time for the multiple block topology was 160 hours. The break down of the time required for elements of the grid generation process for each topology is shown in figure 17. In terms of a 40 hour work week, the single block volume grid would be deliverable 2 weeks earlier than the multiple block grid.

HL-20

As a final illustration of single block grid construction, a grid for the Langley Lifting Body (HL-20), shown in figure 25, was generated. Utilizing the subface decomposition method, the surface of this geometry was sectioned as illustrated in figure 26. The areas of most interest in the CFD computation were the canopy region, the chine regions and the windside of the fin. Grid refinements for the wall were the most time consuming and extensive because of the need to model suspected separated flow. The defining boundaries for the grid lines emanating from the wall were generated quickly as compared to the wall surface grid. The time required to generate the defining boundaries and block topology was 60 hours.

The volume grid was generated in 6 hours of CRAY-II CPU time, requiring 1000 iterations. Figure 27a shows the grid point movement convergence history for each iteration and figure 27b shows the orthogonality source term convergence history. Figure 28 shows a set of 2 representative cross-sections and are expanded in figures 29 and 30. This volume grid did not require any forcing function modifications, and was solvable using an initial relaxation parameter of .6 followed by a switch to 1.0 after 200 iterations as indicated by the jump in residuals in figures 27a and 27b. The total engineering time required to develop this volume grid was 70 hours.

Conclusions

The data presented shows single block volume grid construction about complex aerodynamic configurations is more efficient than multiple block topologies. The subface decomposition method adds difficulty to the definition of the six faces in a single block volume grid. This obstacle is outweighed by the savings in both user and computer resources. The single block method, enables the volume elliptic grid generator to determine the transitions between complex geometry and the surrounding flowfield domain. By allowing the elliptic solver to determine these transitions, the amount of knowledge required for blocking strategies is minimized which reduces the time required to generate a CFD grid.

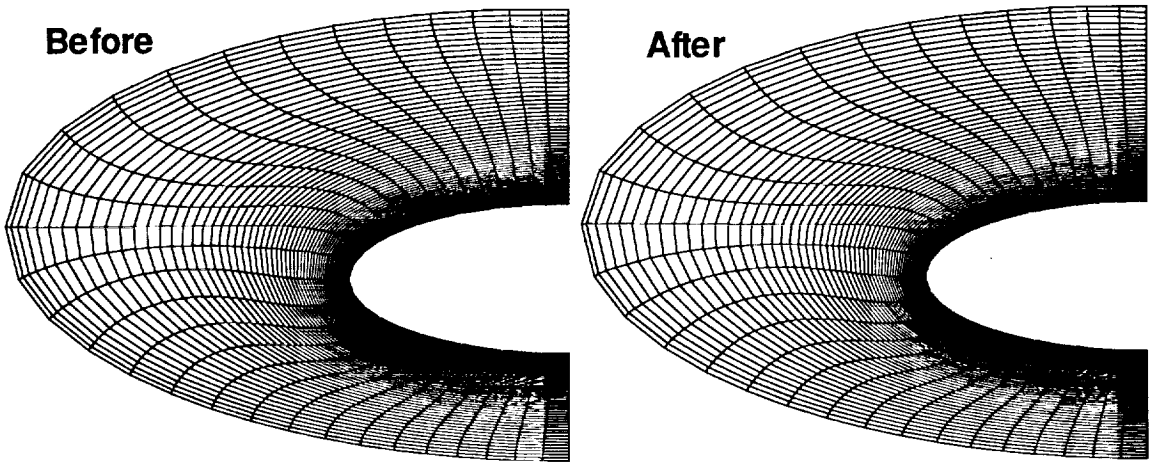
References

[1]Soni, B. K., "Two- and Three-Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics," AIAA Paper 85-1526, 1985.

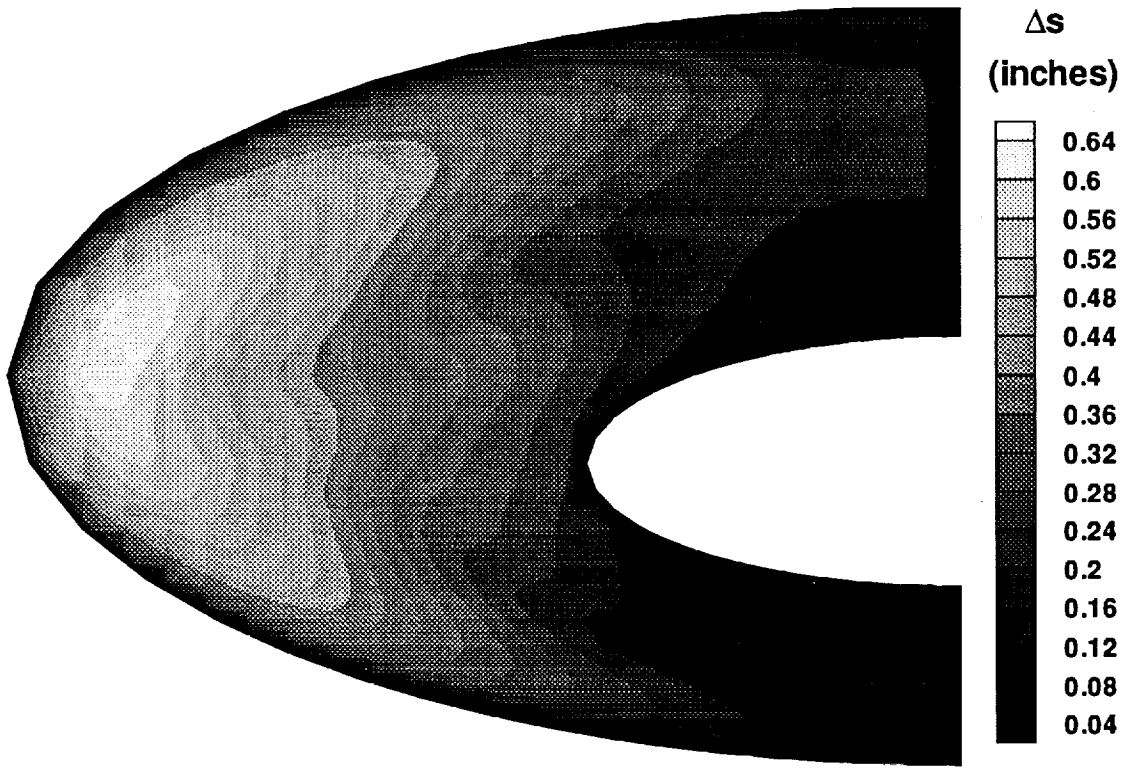
[2]Thompson, J. F., "A Composite Grid Generation Code for General 3D Regions-the EAGLE Code," *AIAA Journal*, vol. 26, pp. 1-10, March 1988.

[3]Steinbrenner, J. P., Chawner, J. R., and Fouts, C. L. "The GRIDGEN 3D Multiple Block Grid Generation System," Wright Research and Development Center Report WRDC-TR-90-3022, October 1989.

[4]Alter, S. J. and Weilmuenster, K. J. "The Three-Dimensional Multi-block Advanced Grid Generation System (3DMAGGS)," NASA TM-108985, April 1993.



(a) Grid distributions on a matching interface before (initial grid) and after (converged solution)



(b) Distance between points of each solution on the matching interface

Figure 1: Differences in grid distributions on a matching boundary surface from before (initial grid) and after (converged solution) running an elliptic 3D volume grid generator.

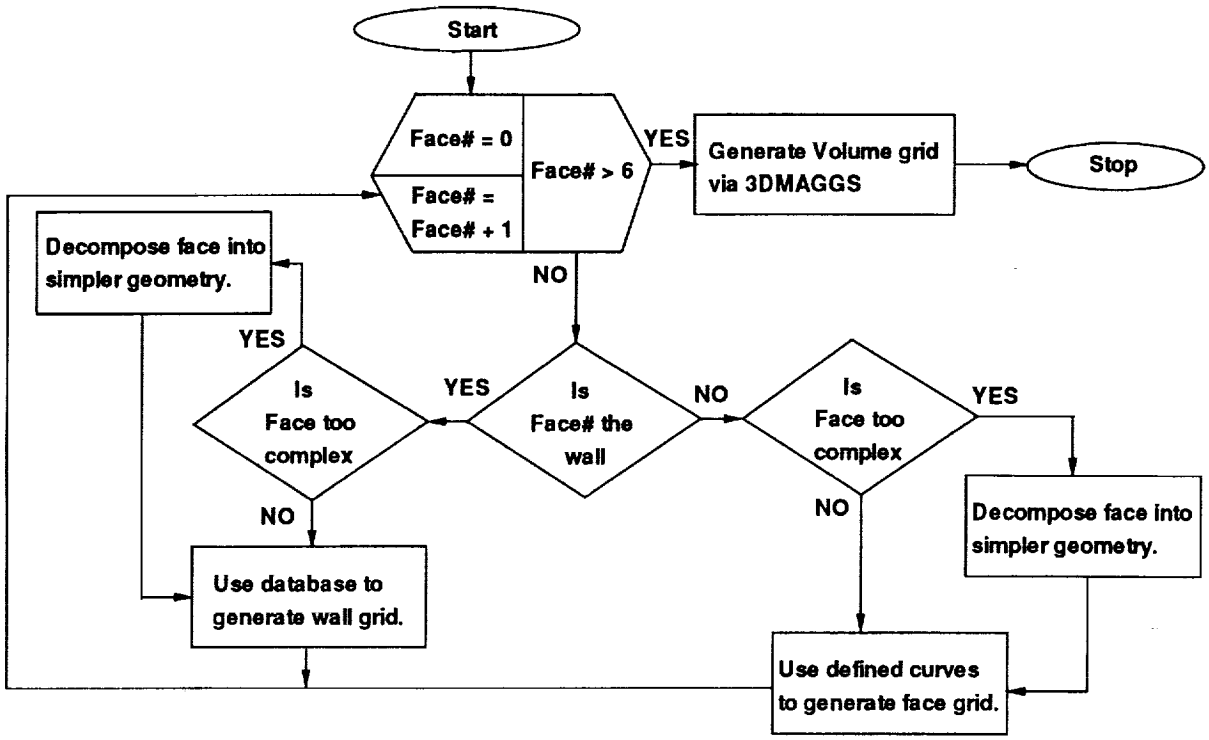


Figure 2: Flowchart identifying methodology for generating the defining grids on any of the six block faces of a single block grid.

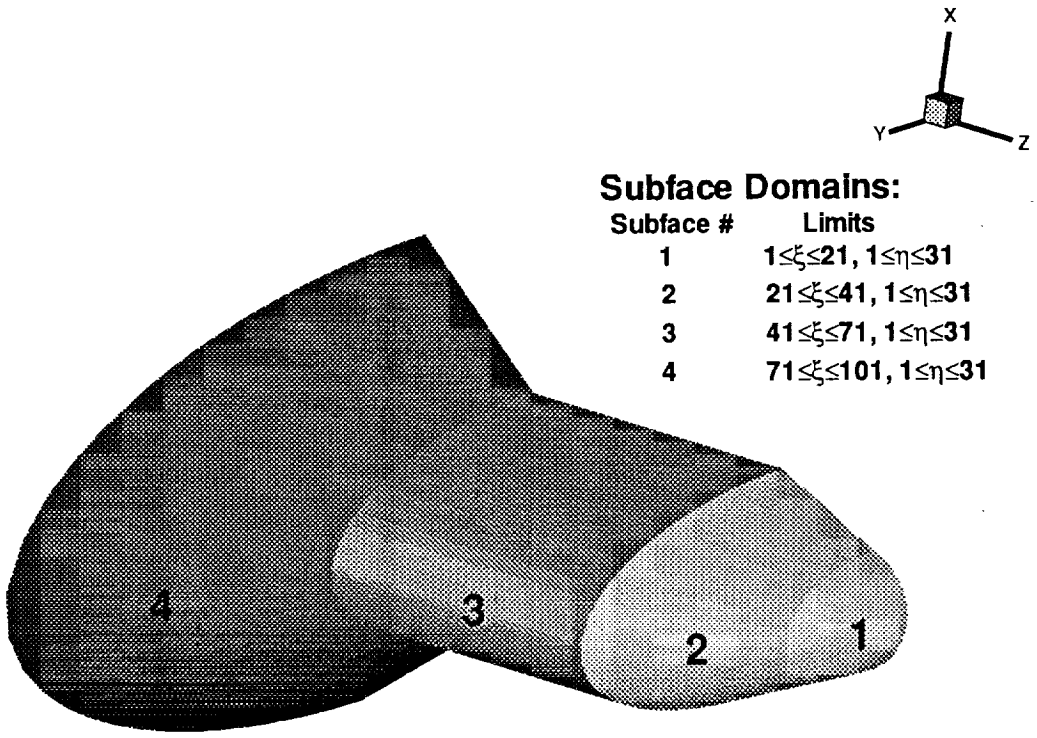
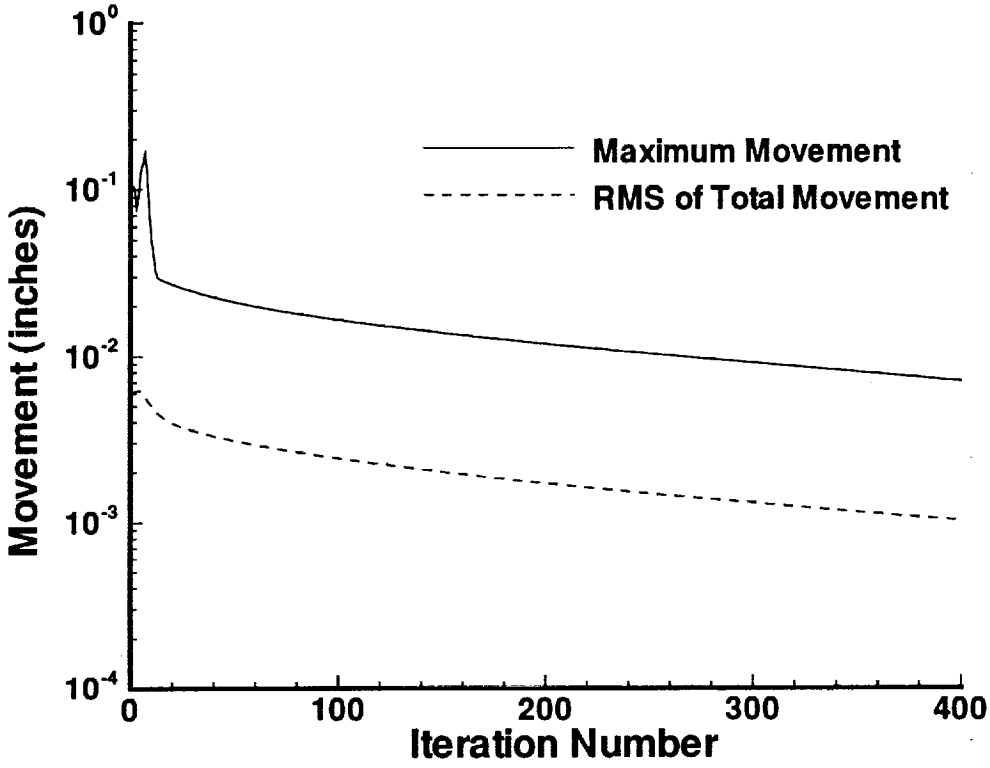
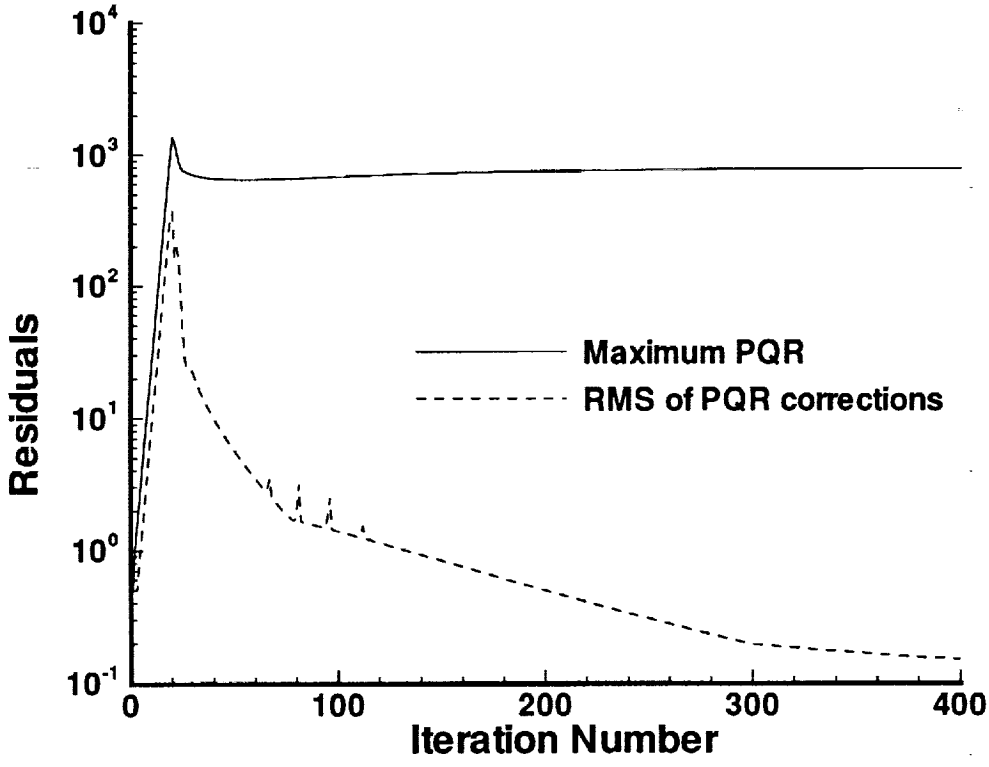


Figure 3: Subface decomposition of the configuration's surface (wall).



(a) RMS residual grid point movement for each iteration.



(b) RMS residual of corrections to orthogonality source termsfor each iteration.

Figure 4: Elliptic solver convergence history for the single block example configuration.

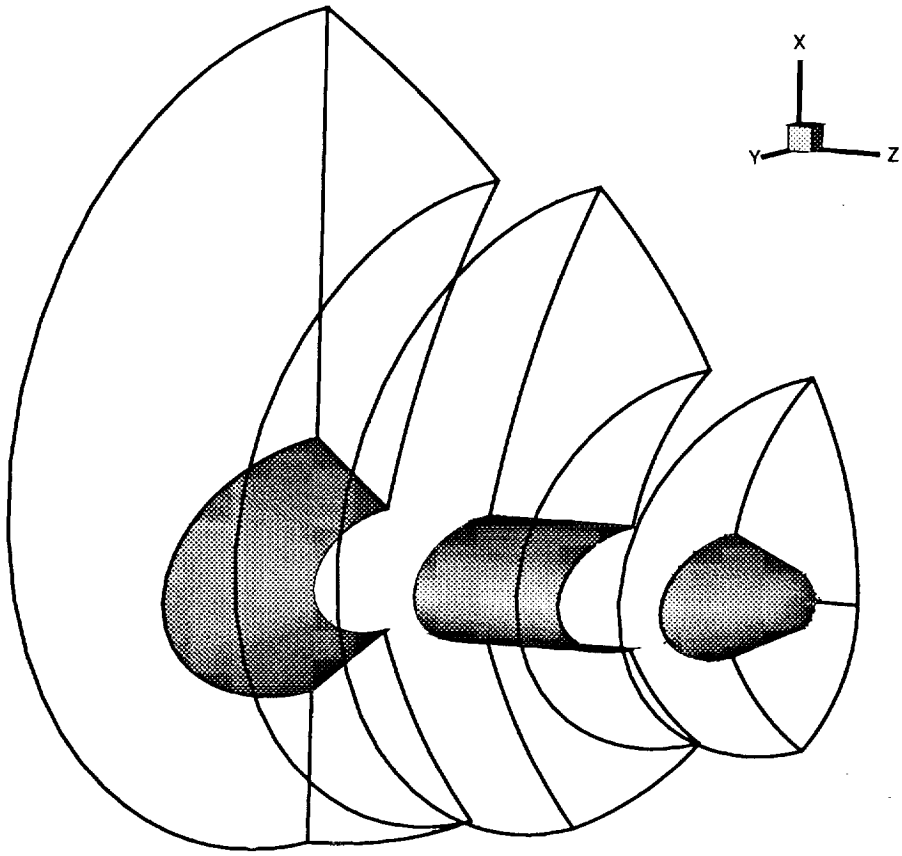
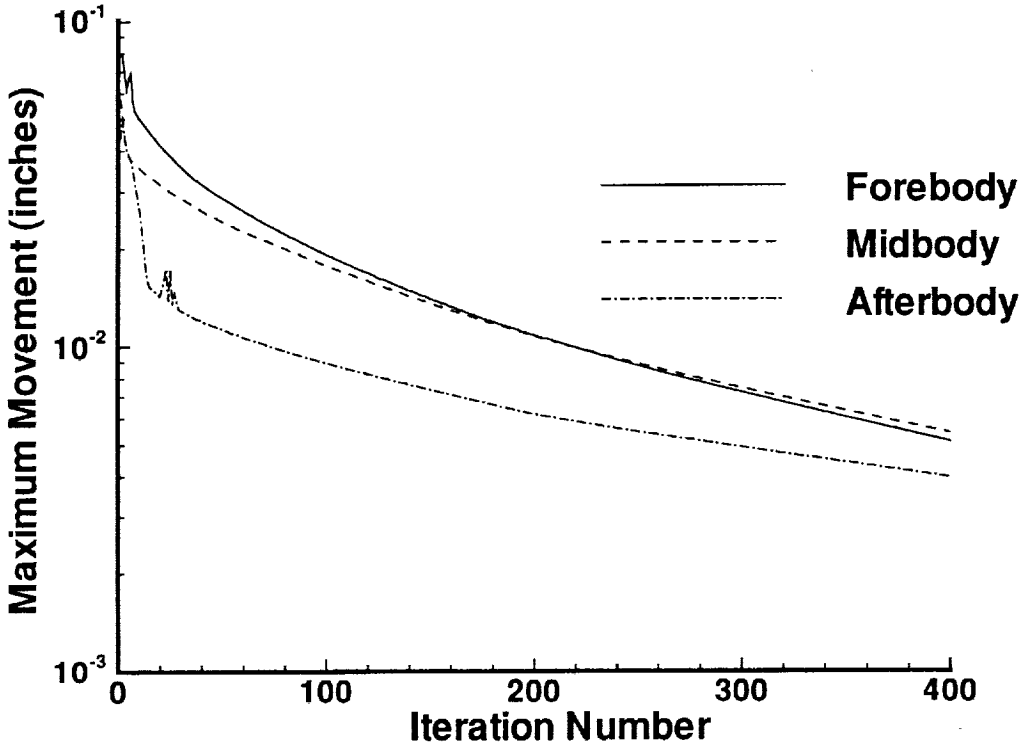
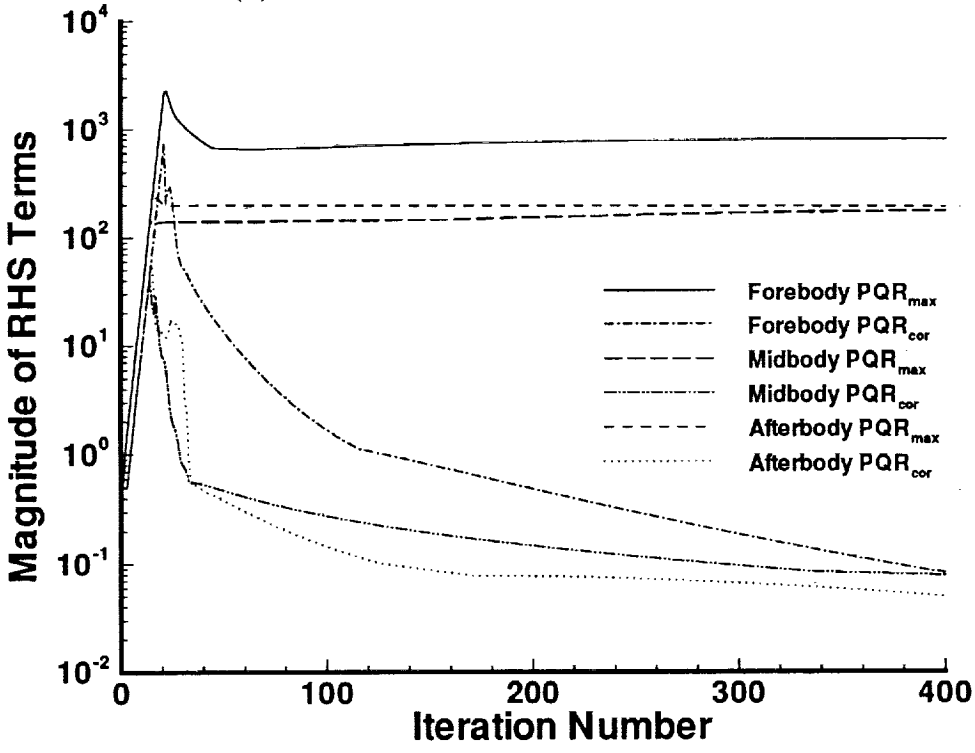


Figure 5: Multiple block decomposition of the example configuration.



(a) Grid point movement convergence history.



(b) Hight Hand Side terms convergence history.

Figure 6: Elliptic solver performance for the multiple block example configuration.

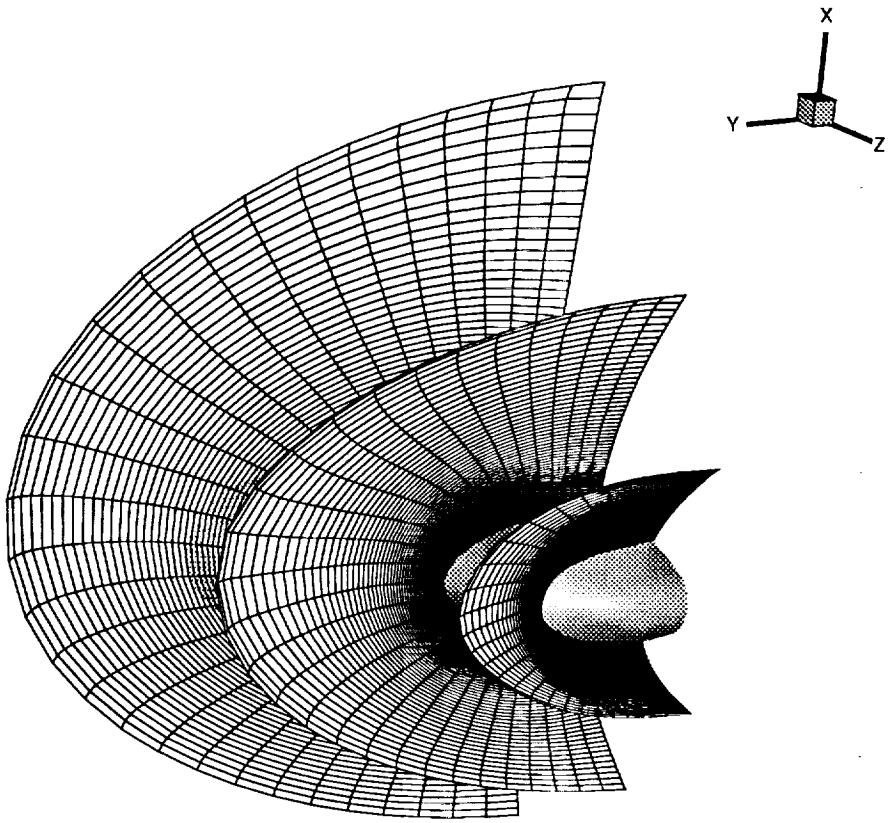


Figure 7: Resulting 3-D volume grid for the multiple block example case.

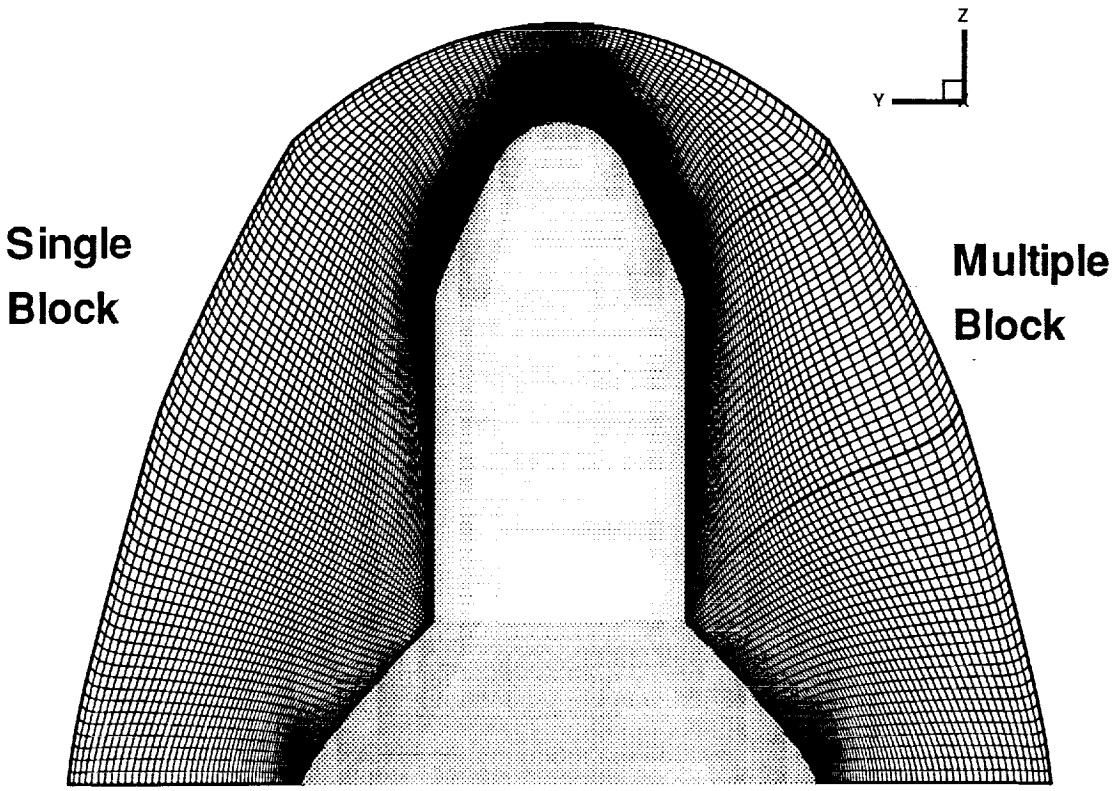


Figure 8: Single block volume grid about the example case comparison to the multiple block topology.

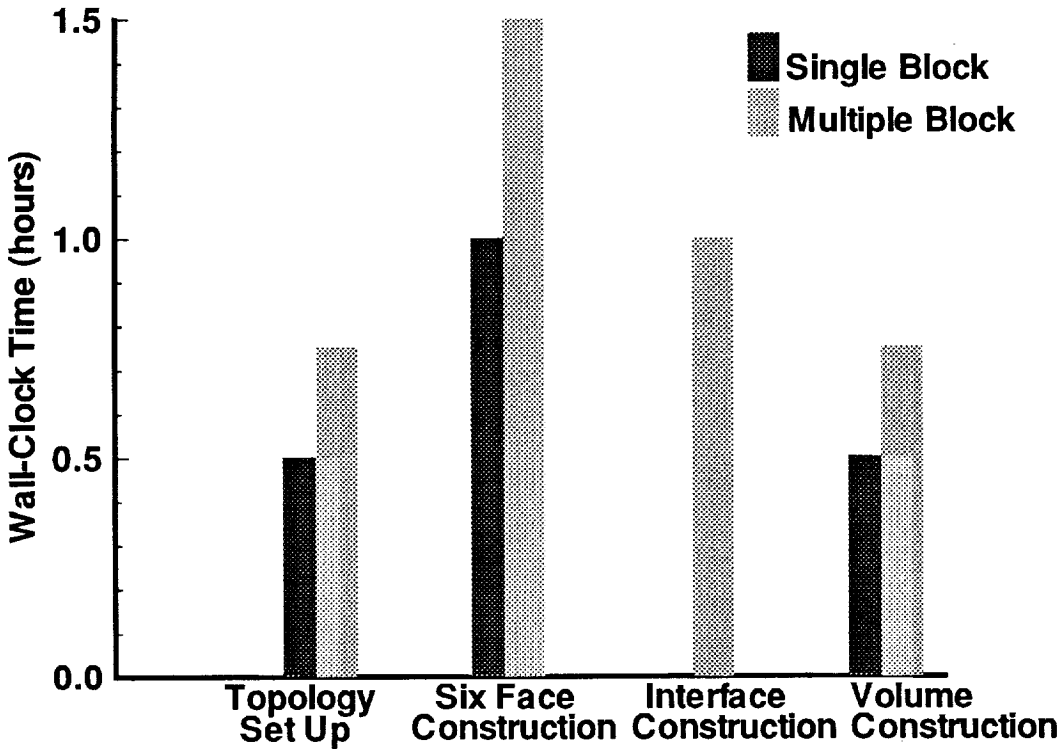


Figure 9: Engineering time to generate single and multiple block topology volume grids.

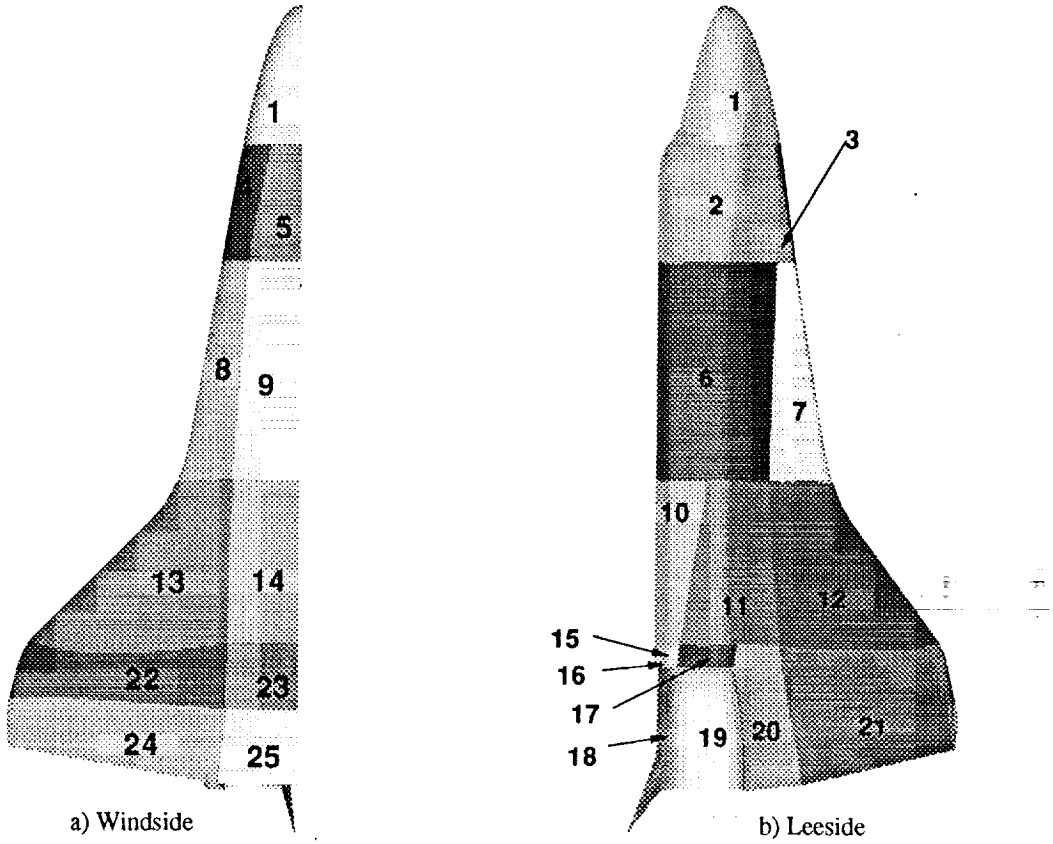


Figure 10: Subface decomposition of the Space Shuttle's surface (wall).

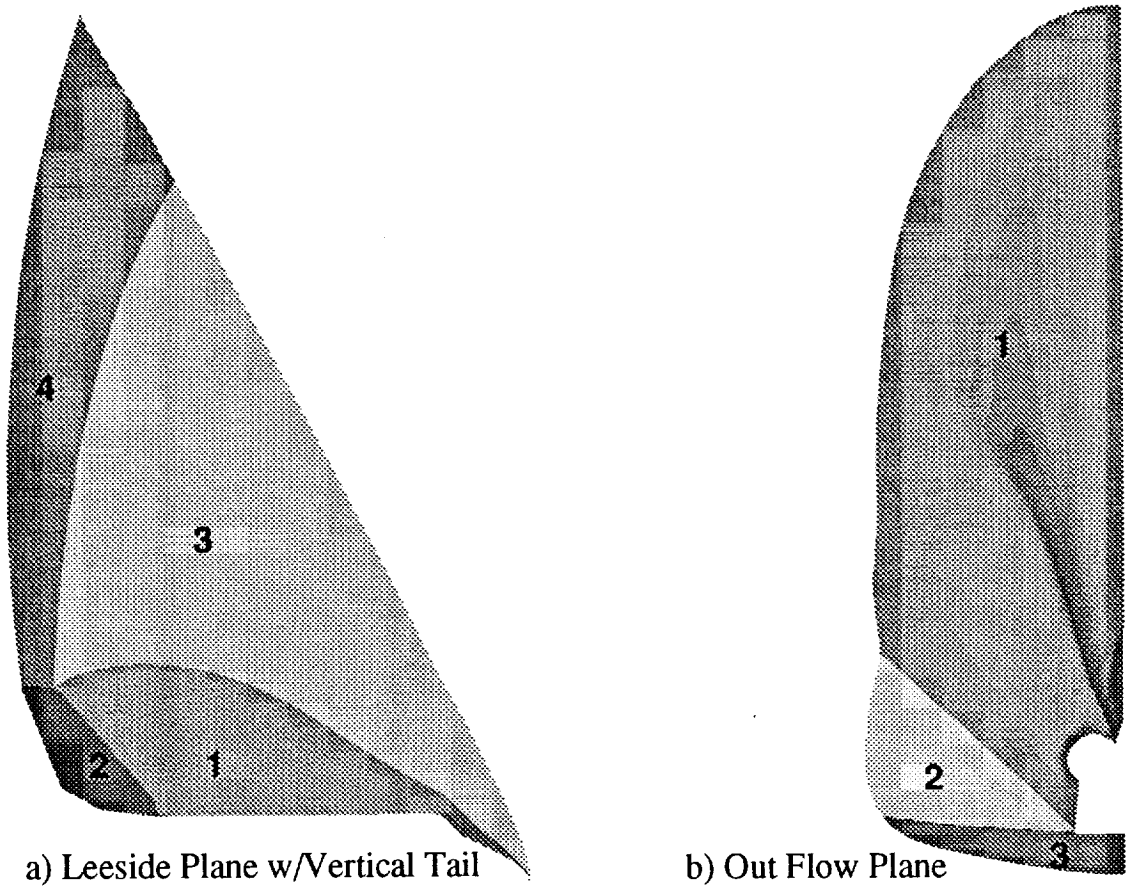
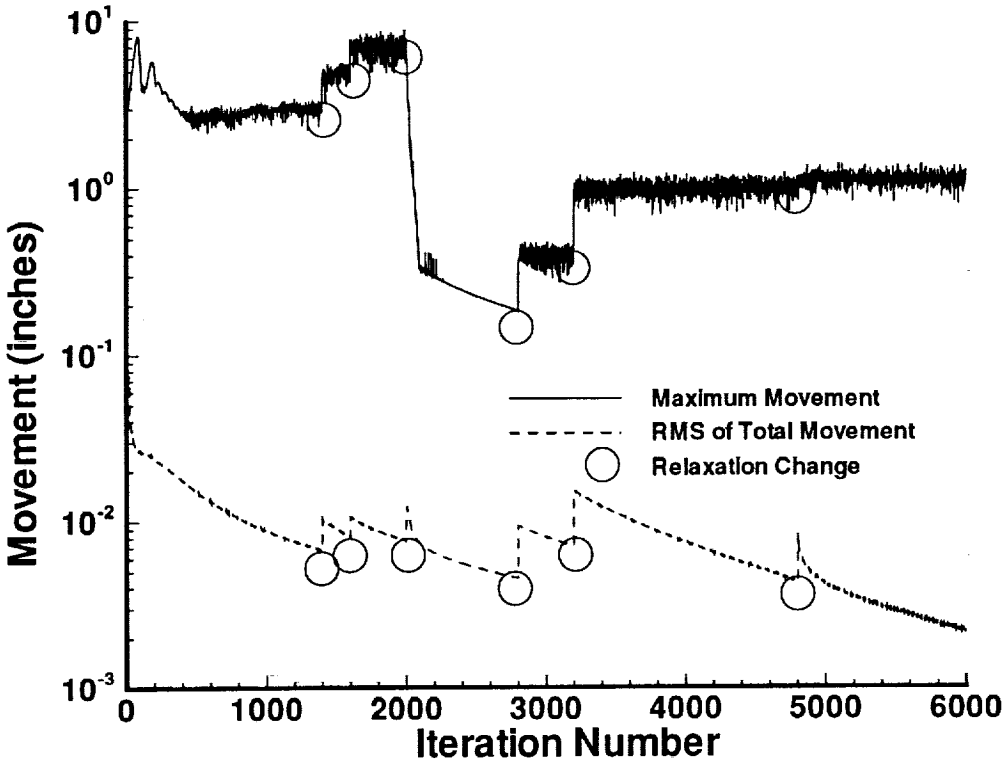
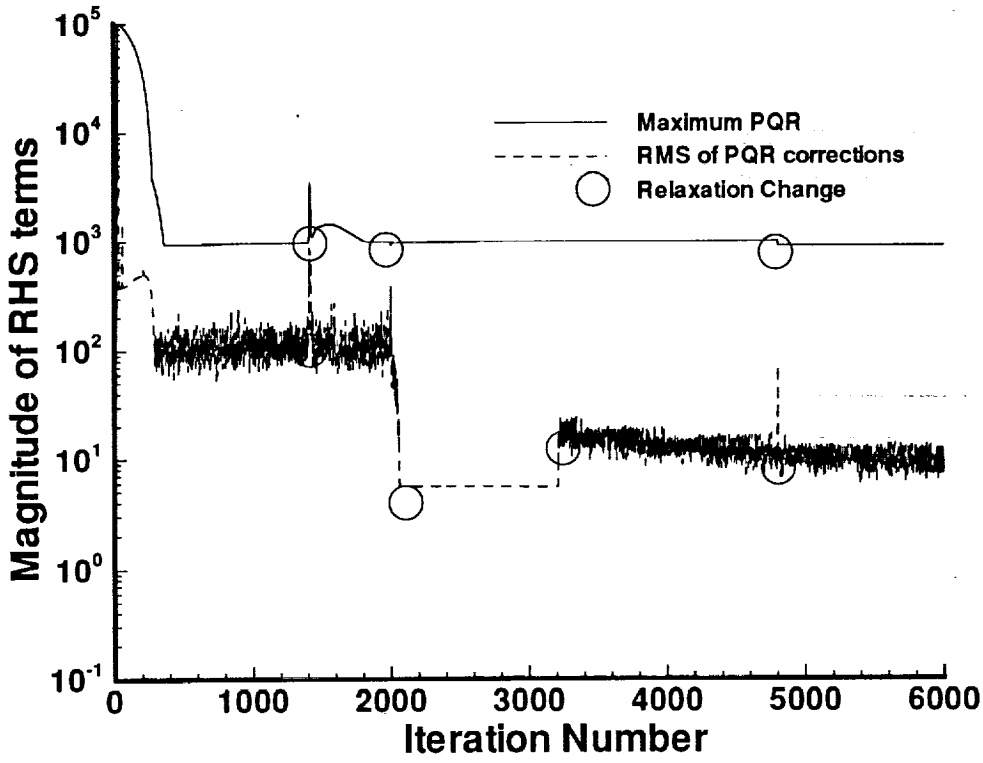


Figure 11: Subface decomposition of the Space Shuttle's leeside and out flow block faces.



(a) Grid point movement convergence history.



(b) Right Hand Side terms convergence history.

Figure 12: Elliptic solver performance for single block Shuttle Orbiter.

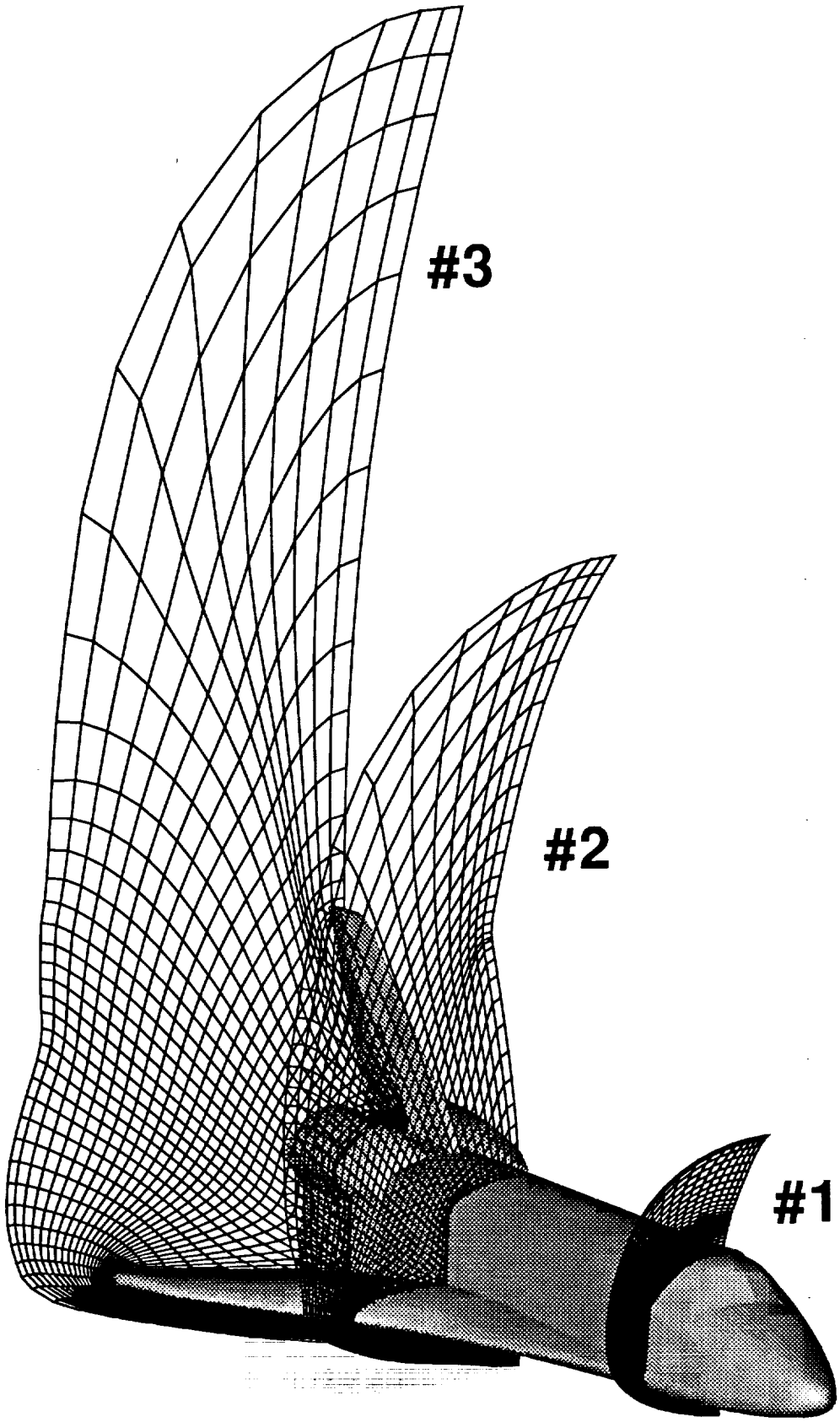


Figure 13: Single block volume grid for Shuttle Orbiter.

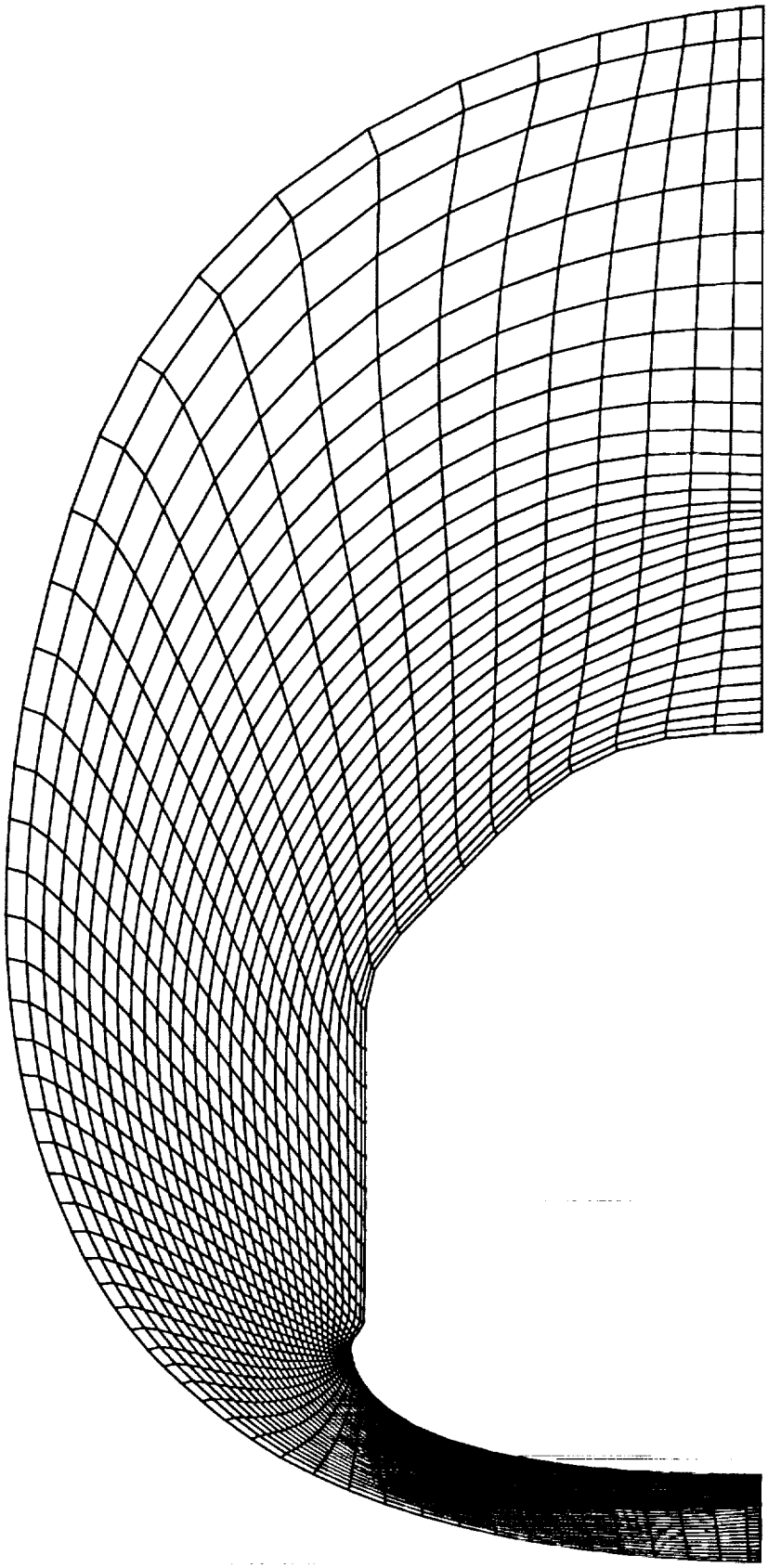


Figure 14: Cross-section #1 of the single block volume grid for Shuttle Orbiter.

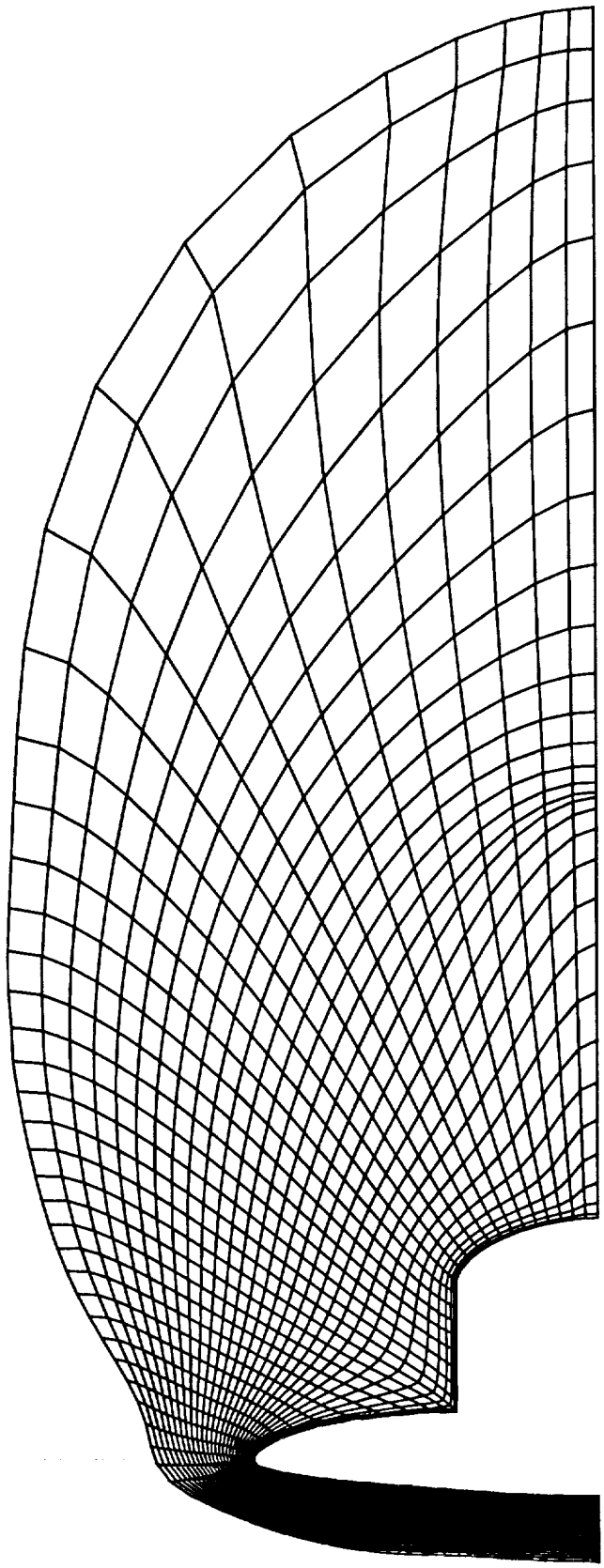


Figure 15: Cross-section #2 of the single block volume grid for Shuttle Orbiter.

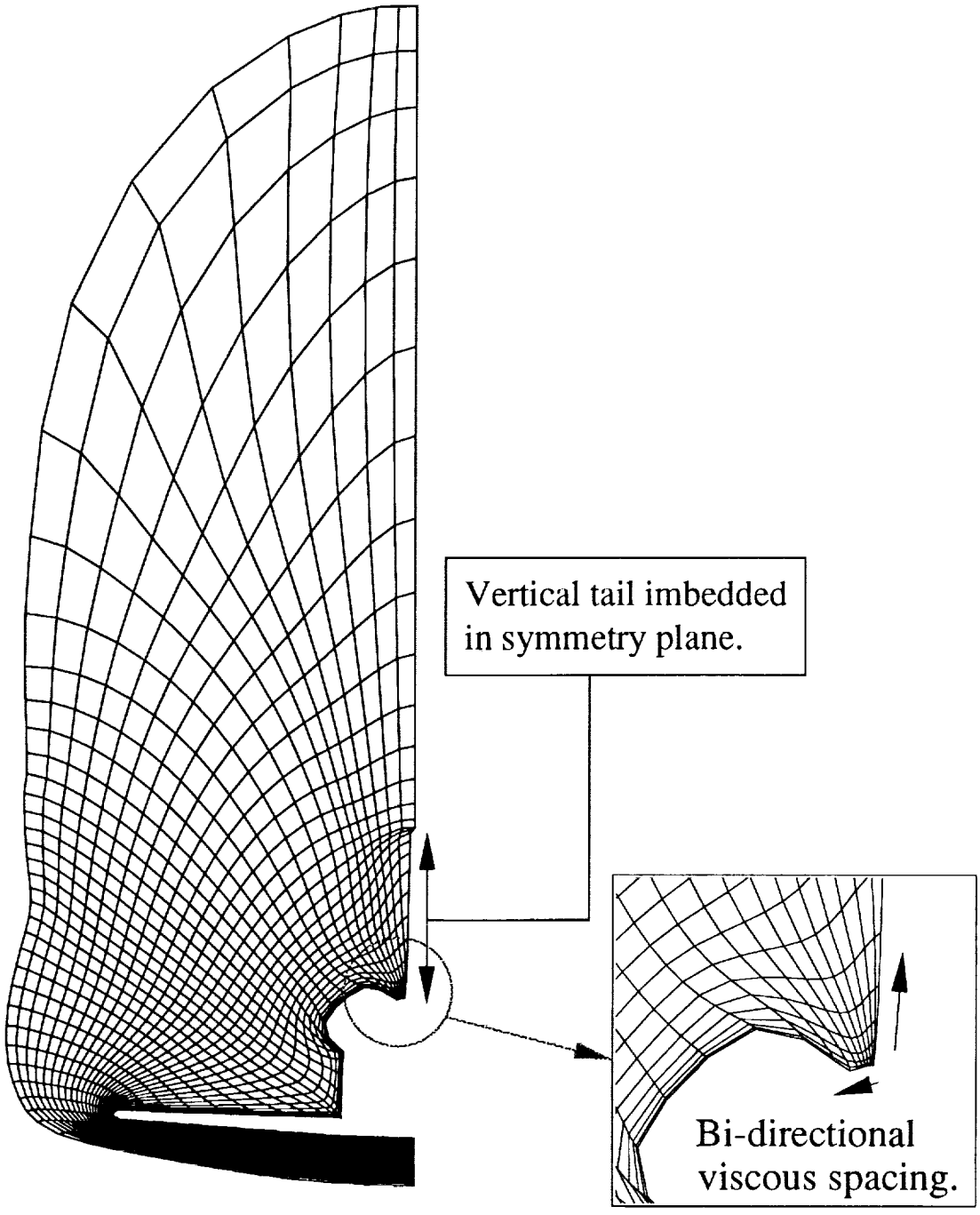


Figure 16: Cross-section #3 of the single block volume grid for Shuttle Orbiter.

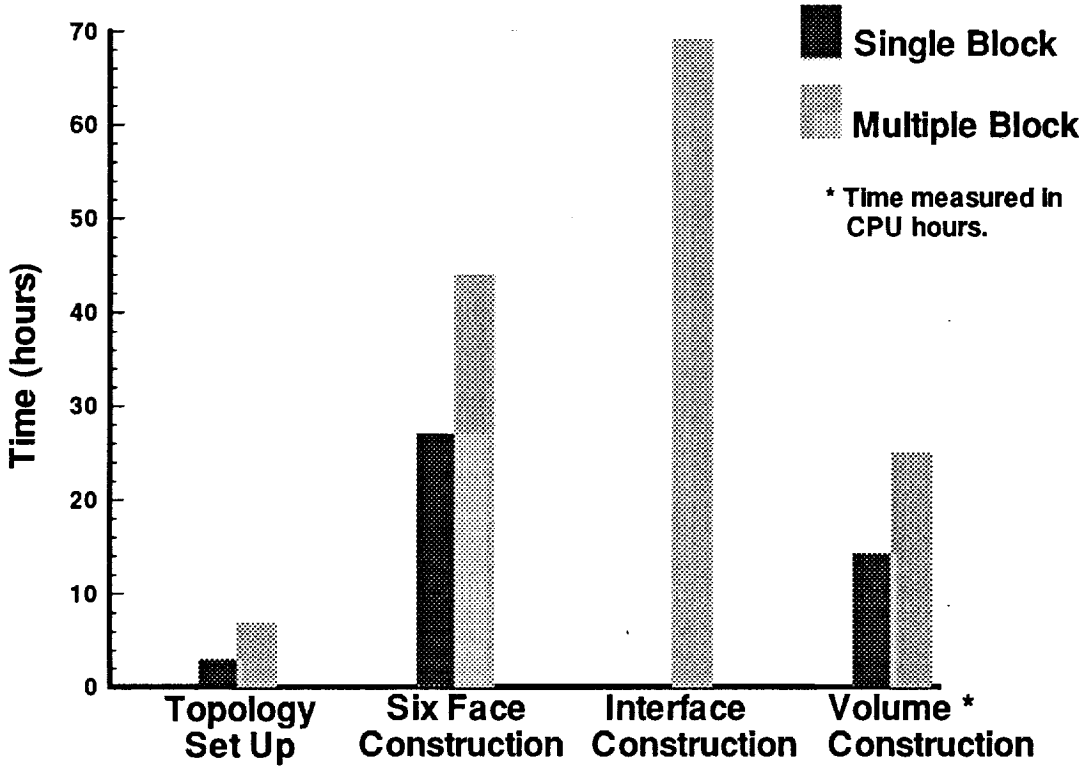


Figure 17: Engineering time requirements for generating the Shuttle Orbiter.

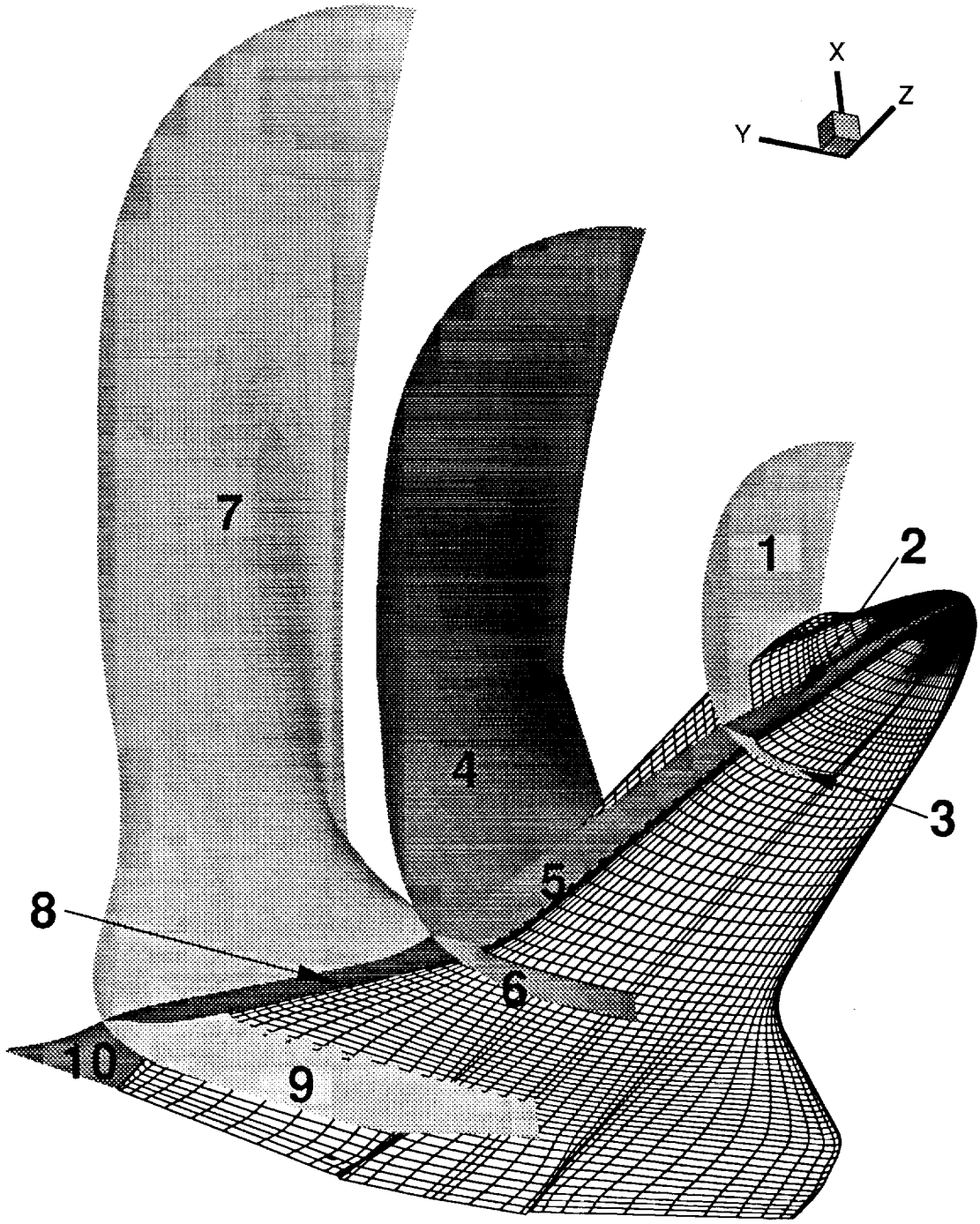
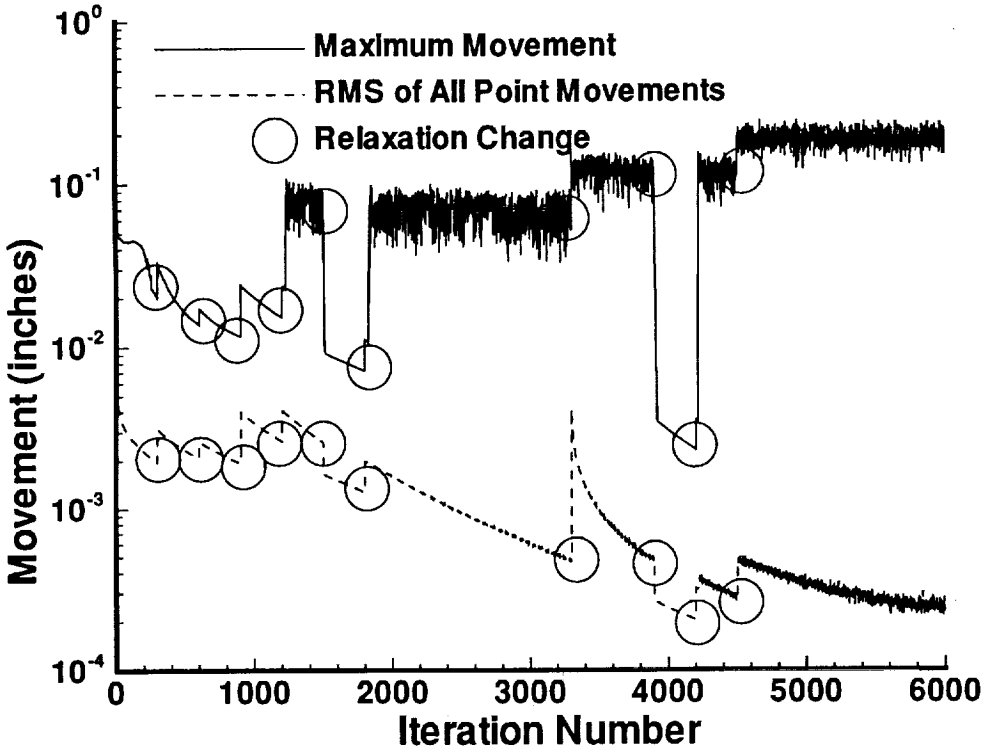
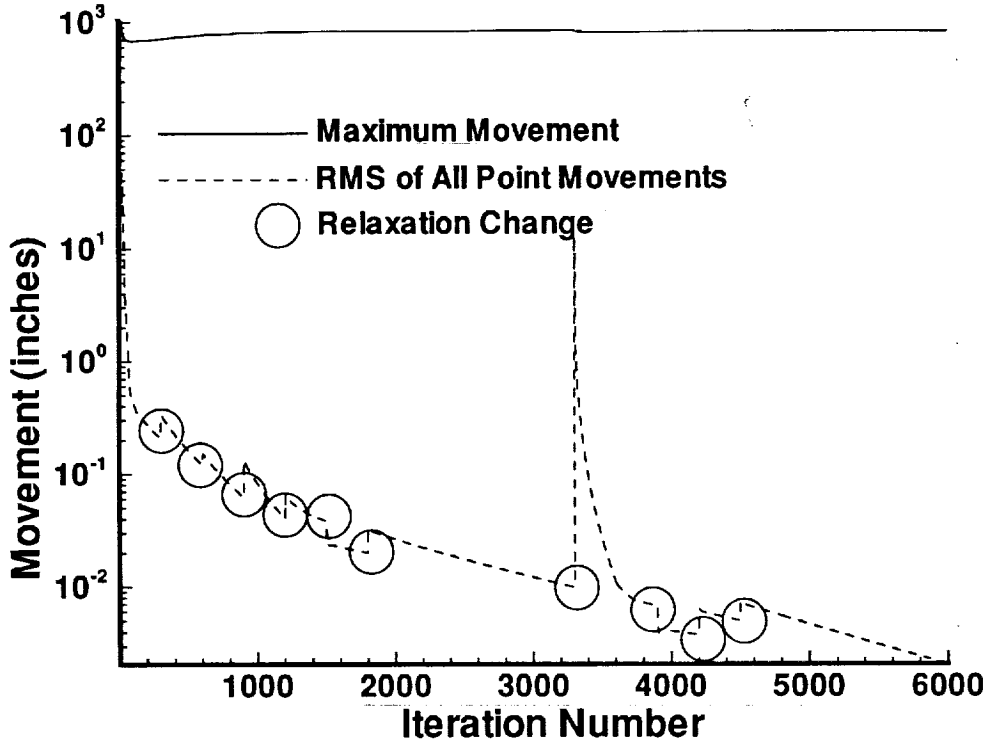


Figure 18: Interface defining surfaces for multiple block Shuttle Orbiter.



(a) Grid point movement convergence history.



(b) Right Hand Side terms convergence history.

Figure 19: Elliptic solver performance for multiple block Shuttle Orbiter.

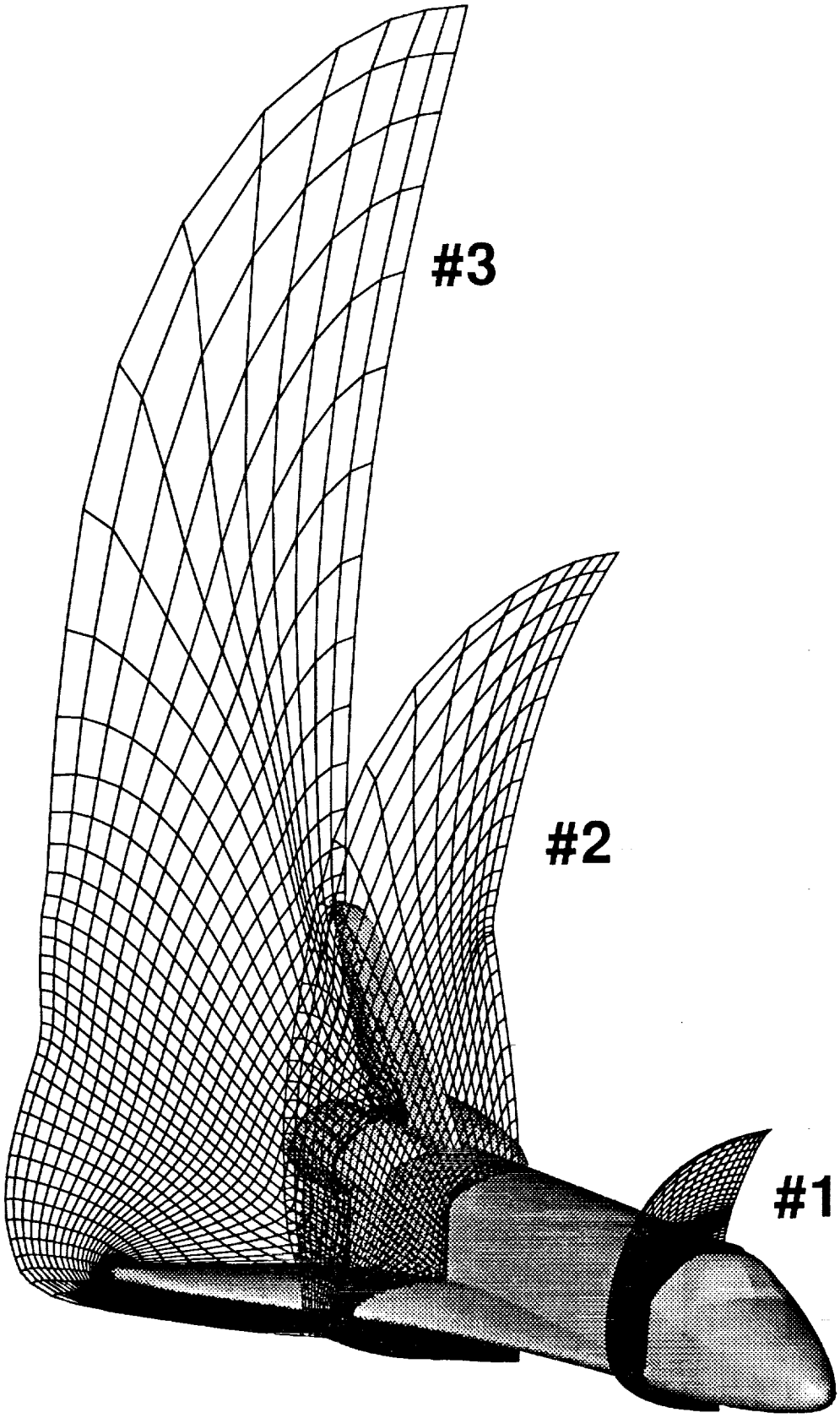


Figure 20: Multiple block volume grid for Shuttle Orbiter.

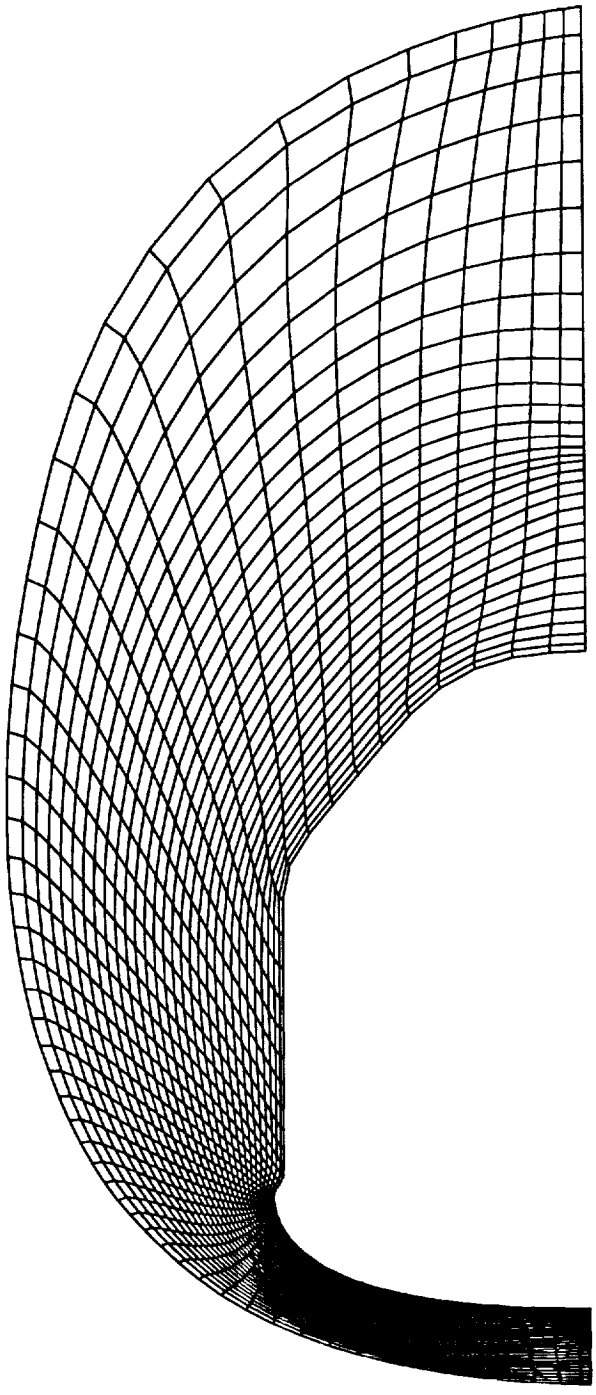


Figure 21: Cross-section #1 of the multiple block volume grid for Shuttle Orbiter.

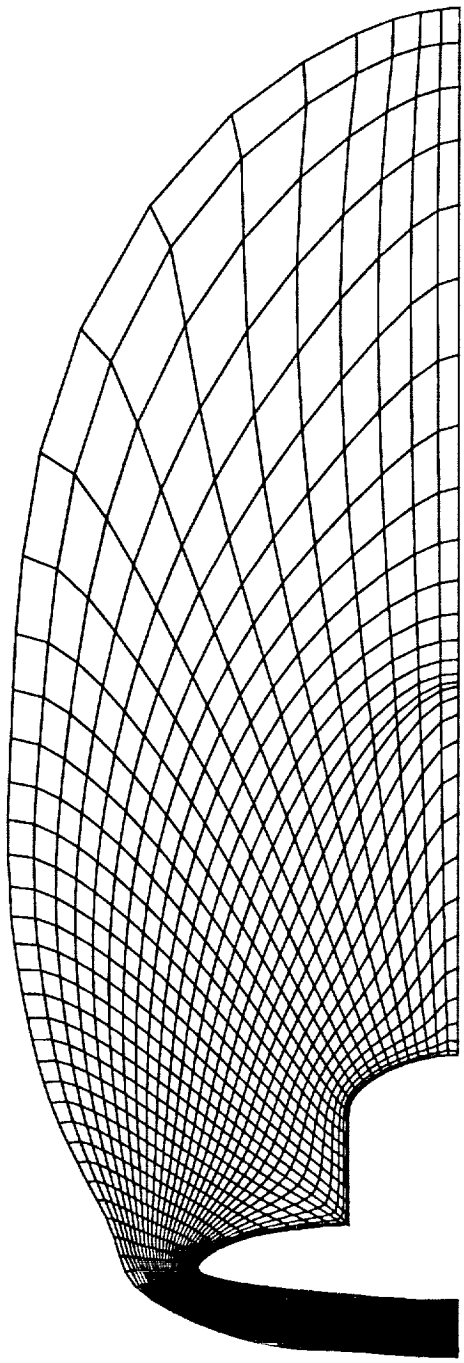


Figure 22: Cross-section #2 of the multiple block volume grid for Shuttle Orbiter.

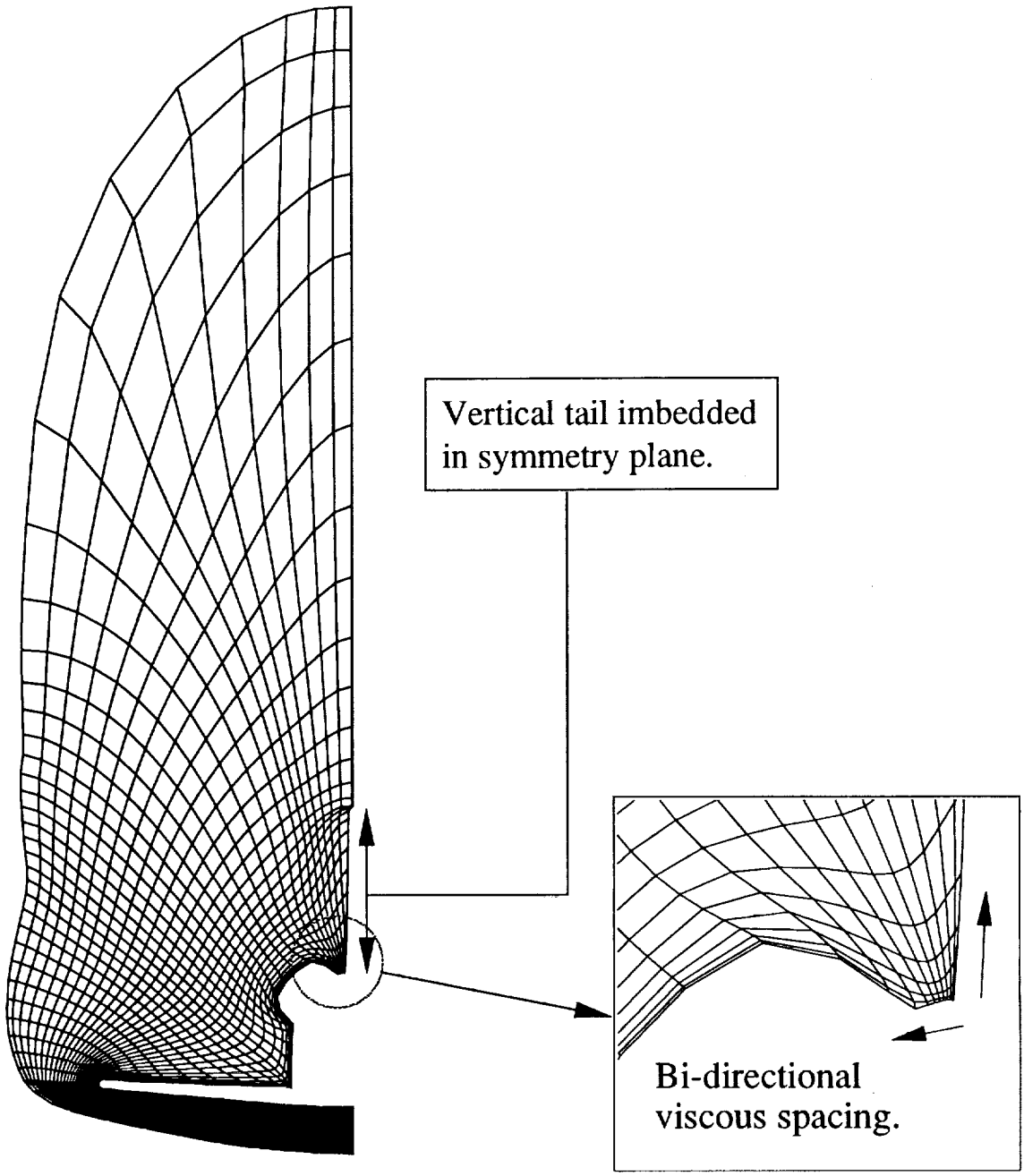
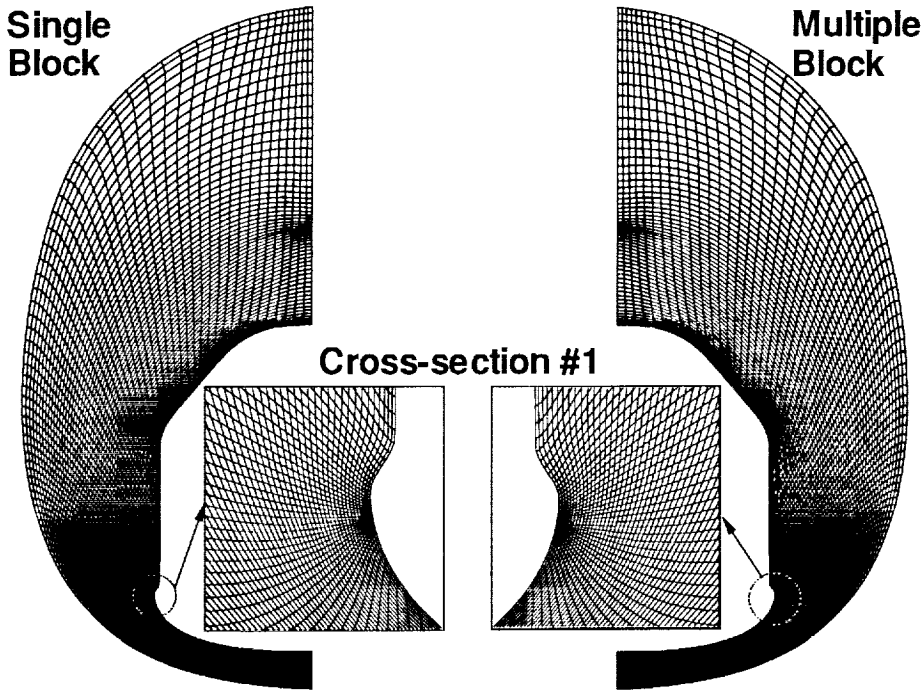
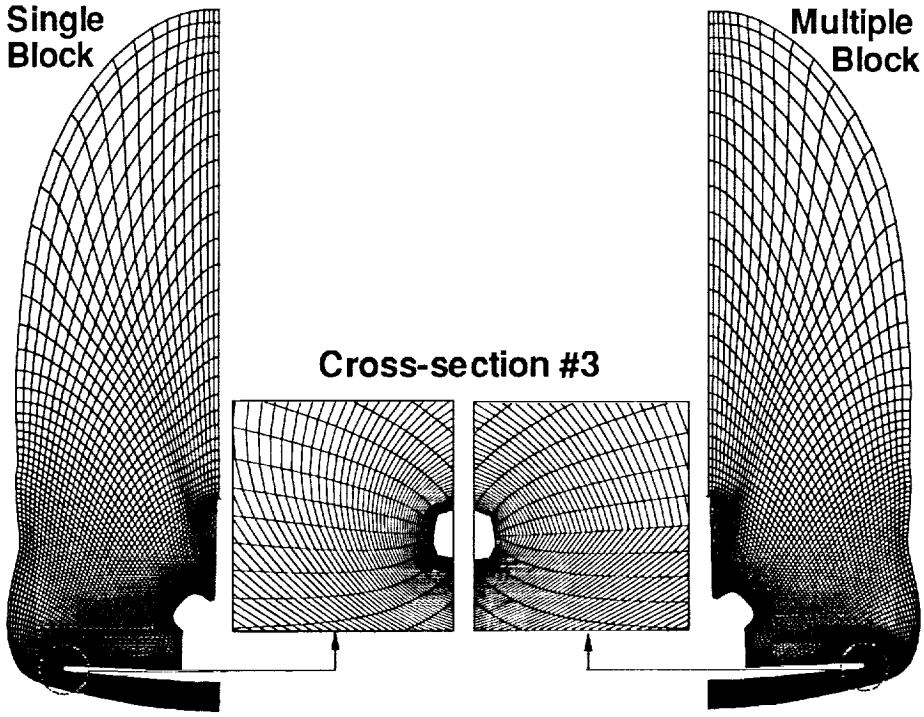


Figure 23: Cross-section #3 of the multiple block volume grid for Shuttle Orbiter.



(a) Cross-section #1 at ξ=40 (900 inches from the nose).



(b) Cross-section #3 at ξ=140 (1200 inches from the nose).

Figure 24: Representative cross-sectional comparisons between the single and multiple block topology volume grids.

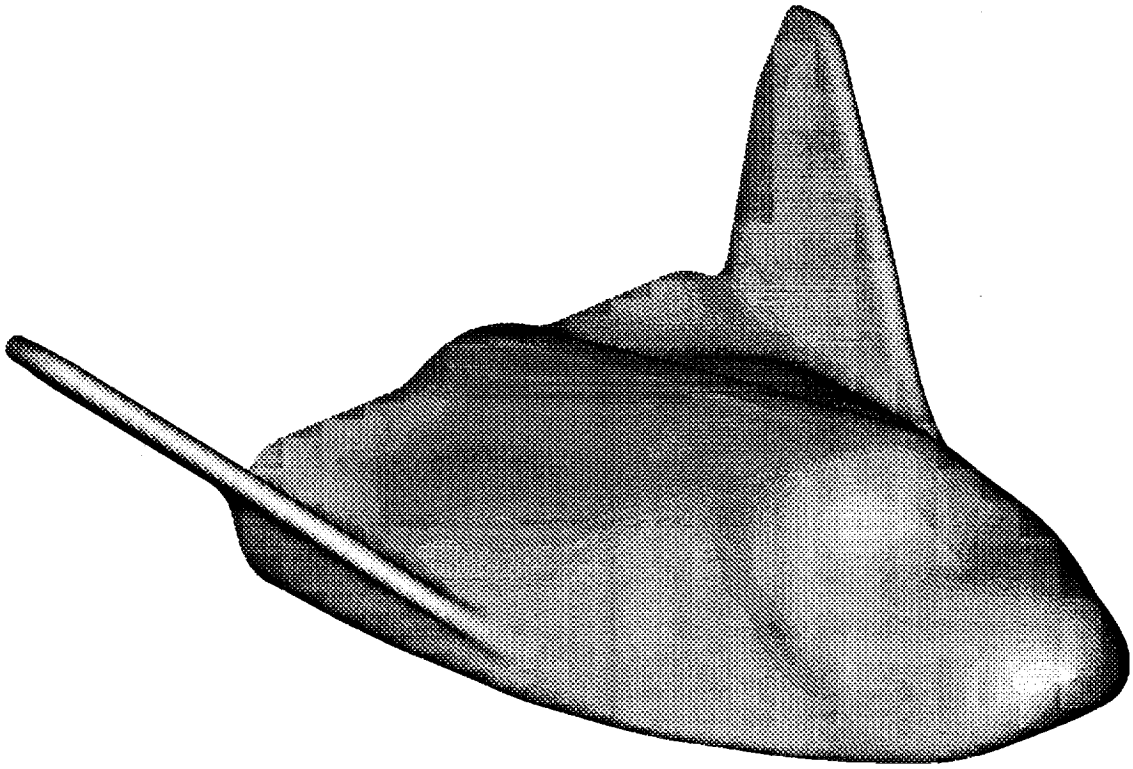


Figure 25: HL-20 configuration wall surface used for volume grid generation.

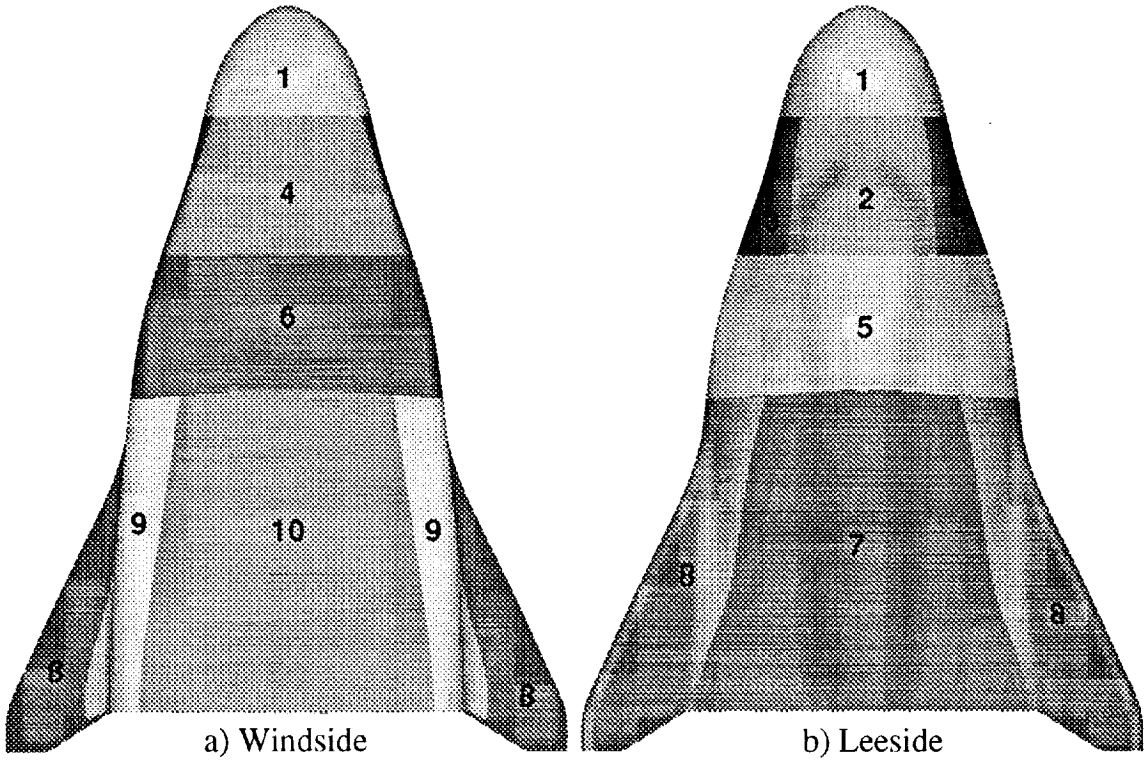
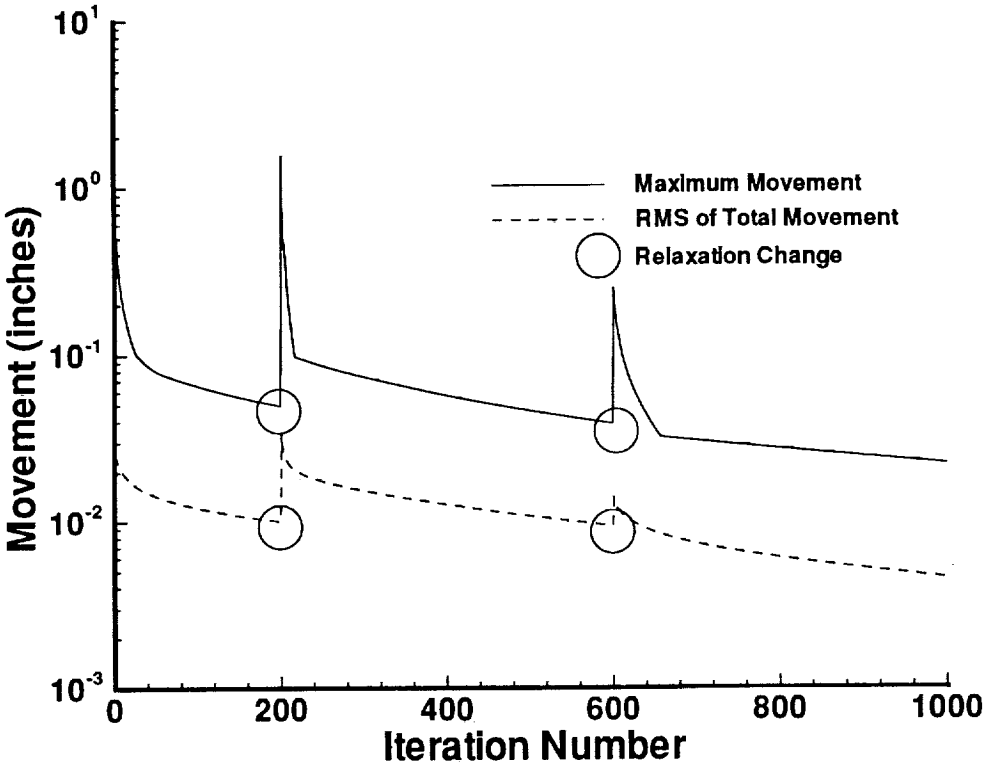
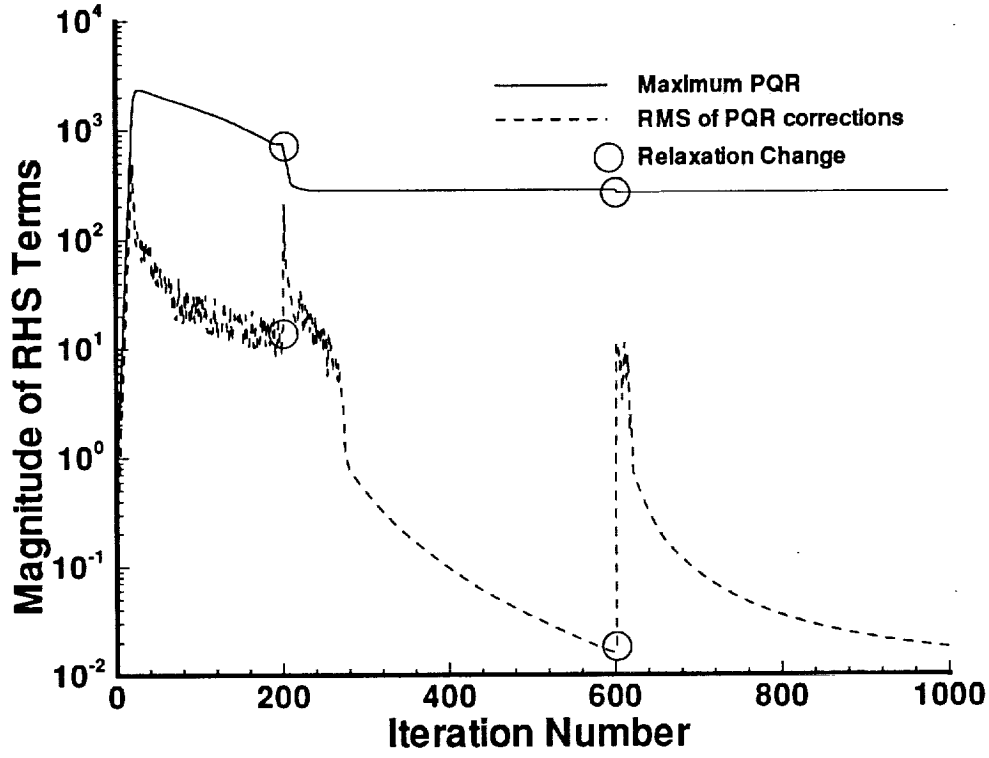


Figure 26: HL-20 Subface decomposition used to construct the vehicle's surface grid.



(a) Grid point movement convergence history.



(b) Right Hand Side terms convergence history.

Figure 27: Elliptic solver performance for the single block HL-20.

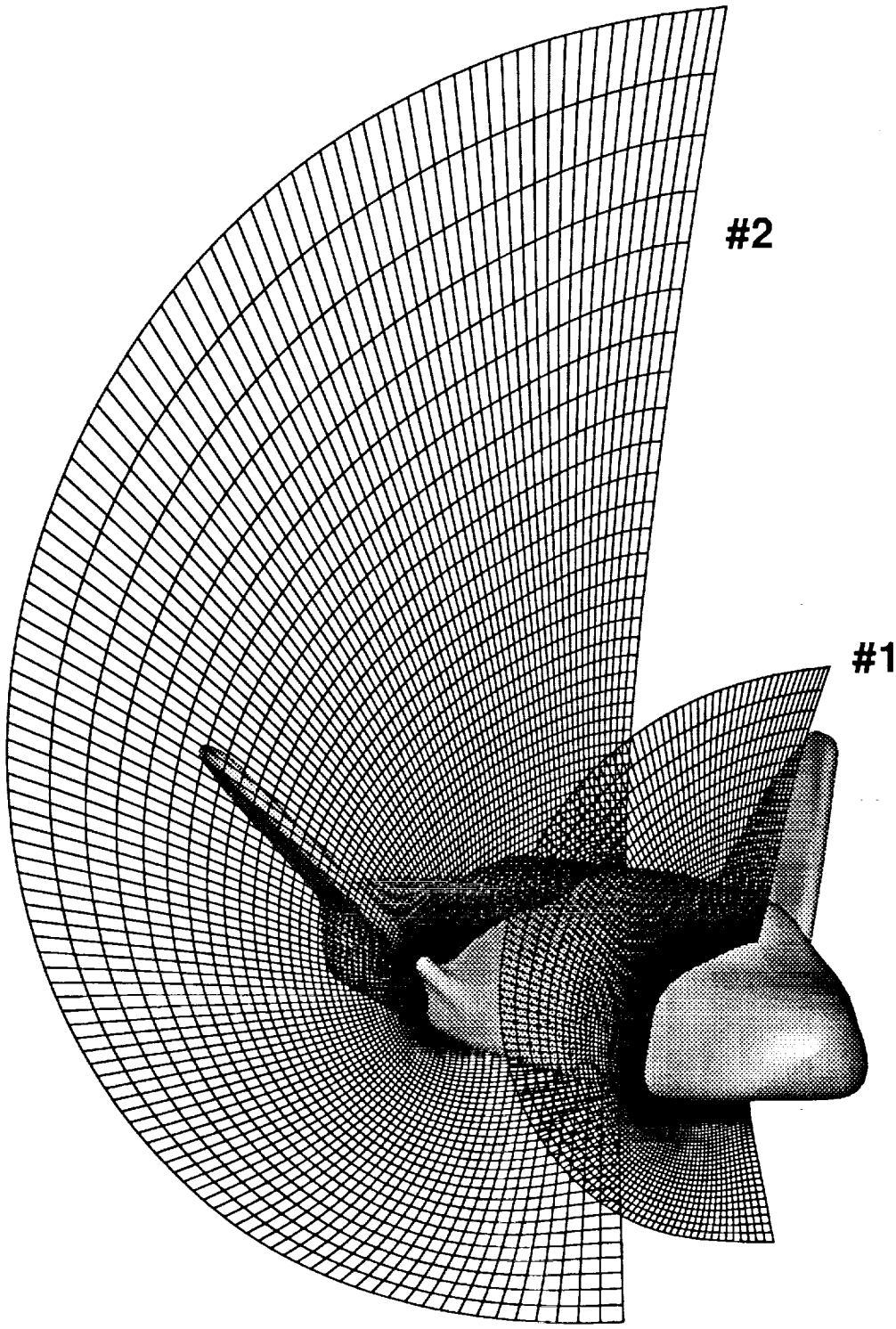


Figure 28: Single block volume grid for HL-20.

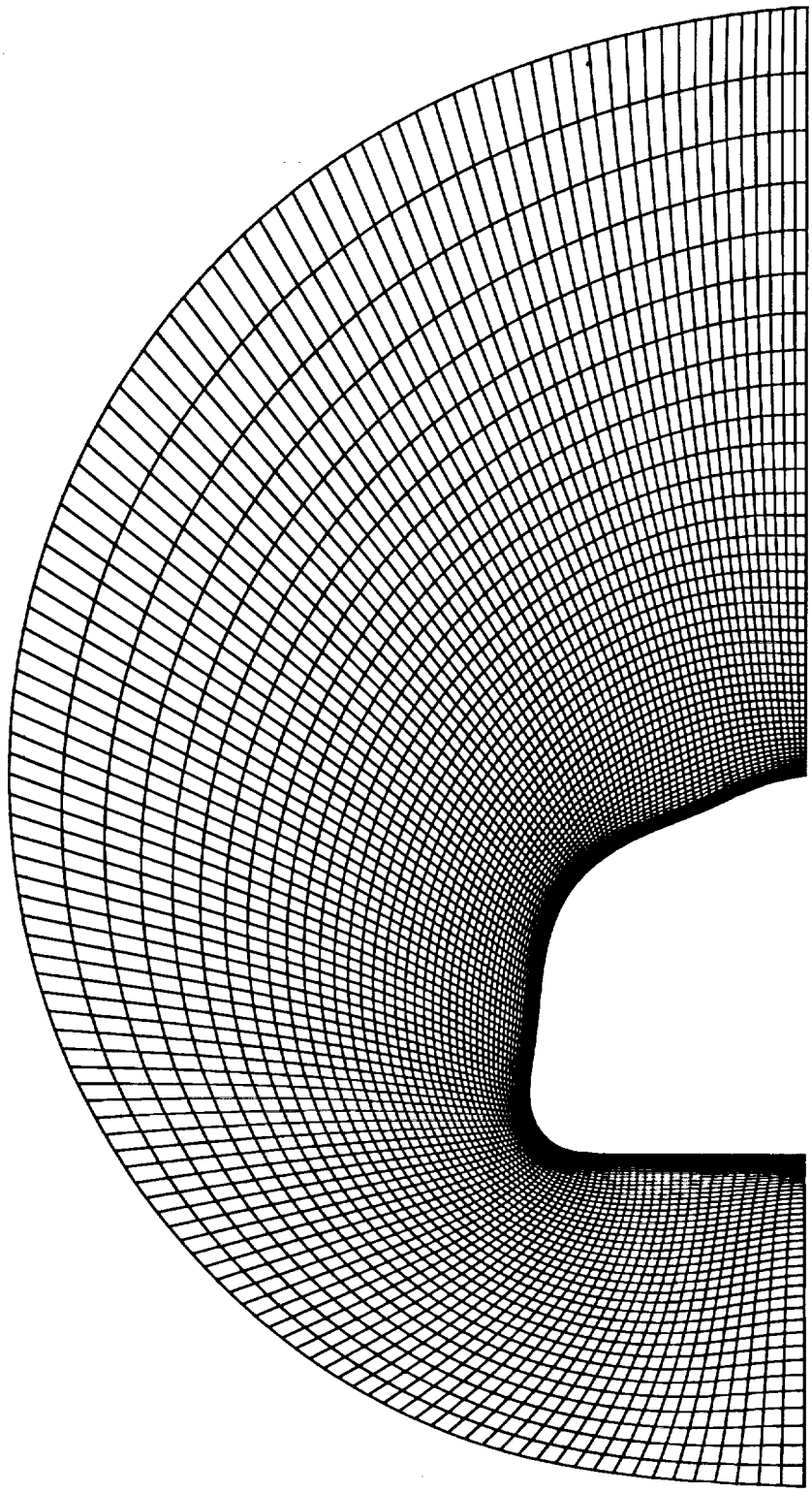


Figure 29: Cross-section #1 of the single block volume grid for HL-20.

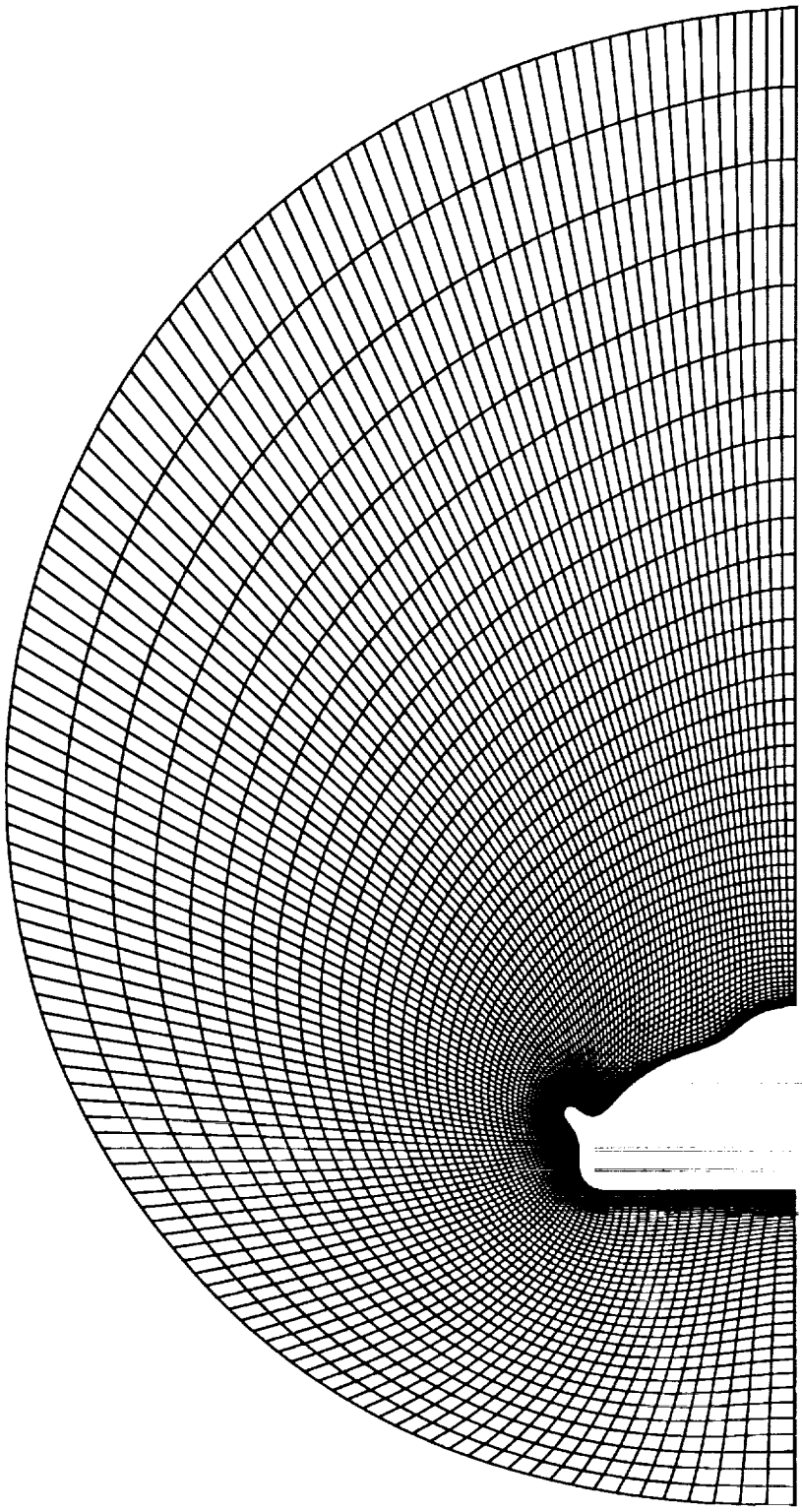


Figure 30: Cross-section #2 of the single block volume grid for HL-20.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1993	Technical Memorandum

4. TITLE AND SUBTITLE
Single Block Three-Dimensional Volume Grids About Complex Aerodynamic Vehicles

5. FUNDING NUMBERS
506-40-91-02

6. AUTHOR(S)
Stephen J. Alter and K. James Weilmuenster

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM-108986

11. SUPPLEMENTARY NOTES
Alter: Lockheed Engineering and Sciences Co., Hampton, VA and Weilmuenster: Langley Research Center, Hampton, VA.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified-Unlimited

Subject Category 34

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This paper presents an alternate approach for the generation of volumetric grids for supersonic and hypersonic flows about complex configurations. The method uses parametric two-dimensional block face grid definition within the frame work of GRIDGEN2D. The incorporation of face decomposition reduces complex surfaces to simple shapes. These simple shapes are combined to obtain the final face definition. The advantages of this method include the reduction of overall grid generation time through the use of vectorized computer code, the elimination of the need to generate matching block faces, and the implementation of simplified boundary conditions. A simple axisymmetric grid is used to illustrate this method. In addition, volume grids for two complex configurations, the Langley Lifting Body (HL-20) and the Space Shuttle Orbiter are shown.

14. SUBJECT TERMS
Langley Lifting Body (HL-20)
Volumetric Grids
Supersonic and Hypersonic Flows

15. NUMBER OF PAGES
49

16. PRICE CODE
A03

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