

1995/22315

## BLOCK-STRUCTURED GRIDS FOR COMPLEX AERODYNAMIC CONFIGURATIONS: CURRENT STATUS

Veer N. Vatsa, NASA Langley Research Center, Hampton, VA  
Mark D. Sanetrik, AS&M, Hampton, VA  
Edward B. Parlette, ViGYAN Inc., Hampton, VA

### SUMMARY

The status of CFD methods based on the use of block-structured grids for analyzing viscous flows over complex configurations is examined. The objective of the present study is to make a realistic assessment of the usability of such grids for routine computations typically encountered in the aerospace industry. It is recognized at the very outset that the total turnaround time, from the moment the configuration is identified until the computational results have been obtained and postprocessed, is more important than just the computational time. Pertinent examples will be cited to demonstrate the feasibility of solving flow over practical configurations of current interest on block-structured grids.

### INTRODUCTION

The field of computational fluid dynamics (CFD) is rapidly approaching the stage where simple configurations, such as wing-body configurations, are routinely analyzed and designed at cruise conditions with CFD tools. However, the situation is not as promising when one considers a more complete configuration, especially at take-off and landing, where high-lift devices must be deployed. The difficulties arise mainly because the task of generating the grids for modeling such complex geometries is tedious, and because the computing time associated with obtaining flow solutions on associated grids containing millions of grid points is excessive. To a large degree, the difficulty associated with generating structured grids for complex geometries has made the unstructured-grid approach more attractive. During last few years, unstructured-grid-based methods have been shown to be very flexible and powerful tools for analyzing inviscid flows by solving the Euler equations. For viscous flows, where the solution of Navier-Stokes equations is needed to capture the flow physics, further research is required to make the unstructured codes more efficient.

Structured-grid-based methods offer a viable approach for solving viscous flows over complex configurations, but have not realized their full potential due to difficulties associated with grid generation. The principle advantage of such methods lies in the efficiency of the associated algorithms in obtaining a solution once a grid has been established. The arithmetic operation count to attain a solution with such algorithms has seen steady decline since early days of CFD research. The advancements in raw speed of the computers have further decreased the computational resources required to obtain converged solution to the governing equations. The difficulties encountered in grid generation can be overcome to a large degree by employing the "divide and conquer" philosophy in constructing block-structured grids for analyzing complex configurations, where each block (or set of blocks) focuses on a given component or region of the domain. Recent work on overlapped, block-structured grids, where structured grids are generated around individual components, has demonstrated the applicability of such an approach for analyzing some of the most complex aerodynamic configurations (ref. 1). The principle drawback of this approach is that conservation is not enforced in the overlapped regions in most codes. From the standpoint of conservation and accuracy, it is desirable to enforce point-to-point match ( $C^0$  continuity) across block interfaces. However, such a requirement imposes severe constraints on the grid-generation process. A reasonable compromise can be achieved by accommodating different grid densities at the block interfaces without the overlapping of grid lines. Such an approach is commonly referred to as a patched-grid approach, and can alleviate many of the difficulties associated with the grid-generation process. In the present paper, we will summarize our experience with the use of block-structured grids for complex configurations.

## OVERVIEW OF CURRENT CAPABILITY

This section is divided into two parts that deal with grid-generation and flow-solver related issues, respectively. The following discussions are kept brief intentionally, because more detailed information on these topics can easily be found in the open literature.

### Grid generation

There is no dearth of block-structured grid-generation schemes for complex configurations, as is evident by the large number of participants in recent conferences on grid-generation (refs. 2–4). Almost all organizations involved in CFD simulations have some type of in-house grid-generation capability. For example, AGPS at Boeing, MACGS at McDonnell Douglas, UNISG at Rockwell/FIAT, and MBGRID at Canadair, are some of the better known grid-generation systems currently in use at various industries. In addition, commercially available software packages, such as GRIDGEN, ICEM/CFD, NGP, and GRIDPRO/AZ3000 are constantly upgraded to meet customer needs. It is difficult to list all such activities, instead we briefly discuss the general features of some of these tools from a CFD user's perspective.

As researchers associated with the CFD Laboratory at NASA Langley Research Center, we are required to analyze flow over a variety of configurations that are of interest to our internal and external customers. Most of the time, these customers provide us with only a coarse point definition of the individual components that comprise the configuration being analyzed. On few occasions, a standard CAD definition ( e.g. in IGES format) may be available. Our task as CFD engineers is to provide accurate flow solutions over such configurations in the shortest possible time.

A suitable grid must be generated before the flow solutions can be computed for a given configuration. The grid-generation task generally turns out to be the most time-consuming part of the process, particularly in terms of man-hours. Given enough lead time, most of the commercial grid-generation systems cited earlier in this paper can be used to create good quality grids, especially by their originators. However, these software packages are quite complex to use and invariably require an expert user (who probably does grid generation for a living) to create good quality grids in a timely manner. The fact that these tools are heavily interactive, with a large number of steps and paths that are traveled via complicated menu-driven branches, keeps the novices at bay. This difficulty can be overcome by creating a center of expertise in an organization, but then one has to face the consequences of compartmentalizing one facet of the solution process from the rest. It is well known that the grid density is determined by the particular algorithm and application, which in turn requires coordination between grid-generation and flow-solution and analysis phases. Such arrangements may not be practical except in large, production-oriented organizations. The point to be emphasized here is that until the grid-generation process is simplified enough to be usable by an average engineer involved in flow simulation activities, CFD as a discipline will not achieve its full potential.

The grid-generation process starts with surface modeling, after the initial geometric definitions become available. These geometric definitions can take many forms, ranging from parametric surface definitions, such as NURBS, to a collection of points measured directly from a physical model. If only a coarse point definition is available, which is generally the case, smooth and accurate surface interpolations are required to create an enriched surface with high fidelity. In general, CAD-based systems, such as ICEM/CFD, have a definite advantage at the surface modeling stage. Usually the surface definition of only the individual components of the configuration (e.g. wing, fuselage, tail ) is provided, and it is left up to the CFD engineer to determine the intersections of the various components. Accurate description of the intersections is essential for maintaining the fidelity of the underlying surfaces. Once again, CAD-based systems are generally more flexible and accurate for computing surface intersections of geometrically complex components. Such flexibility comes at the added cost of acquiring the expertise (through user training) required to navigate through a CAD system, which is generally too complex for average CFD engineer to maintain proficiency. Since surface modeling is only a small part of the total

grid-generation process, a more cost effective way to accomplish such a task is through maintaining a center of expertise, such as GEOLAB at NASA Langley Research Center.

After the surface modeling is completed, the detailed grid-generation process begins by selecting grid topologies for the surface and field grids. For multiblock grids, this requires splitting the physical domain into different blocks (zones), a process which is commonly known as domain decomposition. Most grid-generation packages rely very heavily on interactive, user-supplied intuition to arrive at the blocking strategy (domain decomposition), both at the surface and field grid levels. The interactive approach provides utmost flexibility to the user because it allows full control of blocking strategies. However, such an approach is very tedious, especially for complex three-dimensional configurations, since accurate visualization of the block boundaries in physical space is cumbersome. This is mainly due to the large number of blocks required for gridding complex configurations and due to limitations of available visualization software. To exacerbate matters, especially for  $C^0$  continuous grids, a change in block structure intended for a given zone can cascade into a series of changes affecting several other zones, increasing the effort required to complete the grid-generation task.

Based on our experience, a batch-oriented approach is needed to make the block-structured grid-generation process more attractive to engineering users. The interactive means, especially graphical user interfaces (GUI's) can and should be used to help the user set up the input to the grid-generation code and to help make decisions related to topology and domain decomposition. But, whenever an interactive module is invoked, an easily understandable and editable script file should be created so that it can be added to the batch input file later on. Of course, the user must be able to specify grid density and grid spacings in various zones and subzones. Recently, a batch-oriented grid-generation package with semi-automatic blocking capability was developed by Eisemann and co-workers (ref. 5). The domain decomposition in their software package (known as GRIDPRO/AZ3000) is accomplished through specification of the block structure in a parametric topological space. The zonal boundaries in the physical space move freely and evolve simultaneously with the solution to the partial-differential equations that govern the grid coordinates. The resulting grids are relatively smooth and nearly orthogonal, except near the singular points formed by corners of the grid blocks. Although it is a promising method, in our opinion, several shortcomings of this approach (addressed later in the paper) need to be rectified before this software package can be used routinely by CFD engineers working on aerodynamic flow problems.

Before closing the discussion on grid generation, it should be mentioned that the specification of boundary conditions, especially the block-interface conditions, should be integrated with the grid-generation process. Without such a coupling, too much user time is spent on preparing the input to flow solvers, and the chance of errors increase rapidly with the increase in the number of zones (blocks) and the complexity of the grid topology. This type of capability would also reduce the need for human intervention in linking the grid generation to flow solvers, which would form the core of a complete CFD analysis software system.

## Flow Solvers

During last few years, there has been a steady growth in the number of computer codes capable of solving Reynolds-averaged Navier-Stokes (RANS) equations for steady flows over complex configurations on block-structured grids. Several multiblock codes developed at NASA centers, most notably OVERFLOW, NPARC, CFL3D, INS3D, PAB3D, and TLNS3D have been distributed to various industries, universities, and other federal laboratories throughout the United States. In addition, the large aerospace companies, such as McDonnell Douglas, Boeing, Northrop, Grumman, Rockwell, General Electric, and Pratt & Whitney, have their own customized codes. Researchers at the European Space Agency (ESA), DLR (Germany), and RAE (England) have also developed similar codes for their own internal use. The capabilities of the various codes differ somewhat, depending on the original application for which each code was developed. However, most of the mature codes are capable of handling the complex aerodynamic configurations of practical interest if suitable grids are available.

Before proceeding further with the discussion on flow solvers, it is appropriate to classify different types of block-structured grids, based on the connectivity present at the block (zonal) interfaces. In overlapped (overset)

grids, the grid lines from adjoining blocks form an overlapped region instead of a clearly identifiable zonal interface. In patched grids, the adjoining blocks form a common interface, but the point distribution on the block boundaries that form the interface is different. In  $C^0$  continuous grids, the adjoining blocks that form the interface have a point-to-point match at the zonal interface. The overlapped grids, patched grids, and the  $C^0$  continuous grids form a hierarchy of structured grids requiring increased effort in grid generation; the  $C^0$  continuous grids require the most effort to generate. However, with regard to flow solvers,  $C^0$  continuous grids are the simplest and most convenient to deal with.

The overlapping grids are by far the easiest type of grids to generate; therefore such grids have been used for computations on the most complex geometries to date, such as the flow over the complete space shuttle configuration (ref. 1). However, conservation is not enforced in the overlapped regions of the grid; therefore this methodology may not be appropriate for problems in which accurate capturing of shocks and shear discontinuities is crucial. Furthermore, the actual process of using overlapped grids requires several nontrivial steps beyond the generation of grids around individual components. For example, one must generate collar grids at wing/fuselage juncture regions for adequate resolution of the boundary layers. In addition, a variety of preprocessors are needed to determine the connectivity and interpolation coefficient matrix for the overlapped grids, which requires significant human intervention before CFD analysis can be performed.

The next step in the block-structured grid hierarchy is the patched-grid approach, where the zonal boundaries do not overlap but have different grid distribution in the adjacent zones. Considerable research has been done on patched-grid algorithms to achieve conservation at zonal interfaces. Although a conservative algorithm can be devised more easily for patched interfaces than for overlapped grids, this task is still not trivial, especially for three-dimensional curved interfaces (refs. 6–9). Due to the difficulties encountered in maintaining conservation at zonal interfaces, most general-purpose 3-D flow solvers employ non-conservative formulation at patched interfaces.

For reasons discussed previously, use of the  $C^0$  continuous meshes is preferable whenever possible to guarantee conservation across zonal interfaces. Admittedly, this shifts the burden of CFD simulations to grid generation. We realize that the generation of  $C^0$  continuous grids for complex configurations is a difficult task; however, the advantages of using such grids in terms of global conservation, free-stream preservation, and simplicity of zonal interface boundary conditions can be significant. For example, devising a procedure to check the self-consistency of zonal interfaces in such grids is fairly straightforward. Availability of such diagnostic tools that pinpoint user-input and grid-related errors is extremely helpful to CFD engineers. However, depending on the level of complexity of the geometric configuration, the  $C^0$  constraint on grids may need to be relaxed, especially to make more efficient use of grid points and to improve the overall grid quality (smoothness and orthogonality). By placing the patched-grid interfaces away from strong discontinuities, the errors caused by the loss of conservation at such boundaries can be minimized.

## SELECTED APPLICATIONS

A central-difference, finite-volume, multiblock Navier-Stokes code TLNS3D-MB, developed at NASA Langley Research Center, has been employed to obtain flow solutions on several configurations of practical interest. The thin-layer form of the Navier-Stokes equations is used for modeling the mean flow. Unless stated otherwise, the flow is assumed to be fully turbulent, and the effect of turbulence is modeled through the eddy-viscosity hypothesis. A five-stage Runge-Kutta time-stepping scheme with three evaluations of the artificial dissipation terms computed at the odd-numbered stages is used to advance the flow solution in pseudo time. Implicit residual smoothing is used to increase the stability of the time-stepping scheme. Further enhancement in convergence of the time-stepping scheme is achieved via a multigrid acceleration technique. The details of the numerical algorithm used in TLNS3D-MB are available in references 10–11. This code has been calibrated through a wide variety of applications by several independent researchers (refs. 12–15).

For a flow code to be accepted by the research community at large, it must be capable of providing accurate solutions for problems of interest in a timely manner. To a large degree, the accuracy of a flow solver depends

on the explicit and implicit levels of artificial dissipation inherent in the numerical scheme. Based on our earlier research, it was concluded that the accuracy of the central-difference scheme is greatly enhanced by using a matrix type of artificial dissipation model instead of a scalar dissipation model (ref. 16); hence, the matrix dissipation model has been used for the computations presented here. Another key element that determines the overall accuracy of a CFD code is the ability of the underlying turbulence model to capture the pertinent flow physics. If a flow solver has a turbulence model that fails to accurately predict the important flow features, it will not meet the requirements of CFD engineers involved in the design process. For aerodynamic applications, the ability to predict separation zones, shock locations, and boundary-layer properties in the presence of strong pressure gradients is extremely important. In addition, the turbulence model should be able to accommodate multiple surfaces that intersect one another. Based on both a literature survey and our own experience, the one-equation turbulence model of Spalart-Allmaras (ref. 17) can produce accurate solutions for a wide variety of aerodynamics problems, and has been used in the current applications.

The multiblock structure in this code was constructed carefully to minimize the communication lag between the blocks, thereby achieving convergence levels comparable to the single block implementation. Of course, some penalty in the computational efficiency is inevitable because of the overhead associated with the added complexity of coding required in a multiblock code. It is preferred to use the largest block size subject to the geometric constraints to retain the vectorization efficiency. Such a strategy also enhances the implicitness of the numerical scheme by increasing the domain over which the implicit operator of the residual smoothing is effective. These issues are discussed in detail with specific examples by Vatsa, Sanetrik, and Parlette (ref. 10). In the next section, several applications of this method to problems of general interest are discussed.

### Multi-element airfoil

During landing and takeoff maneuvers, most aircraft deploy a wing configuration that consists of a multiple number of airfoil sections. For structural and aerodynamic reasons, these components are placed extremely close to one another. Such an arrangement creates special problems in the construction of suitable structured grids, especially  $C^0$  continuous grids. An excellent test case representative of realistic high-lift configuration is now available as a result of a joint effort between NASA Langley Research Center and McDonnell Douglas Aircraft Co. for a two-dimensional, 3-element high-lift configuration. The newly available GRIDPRO/AZ3000 software has been used successfully to grid this configuration (ref. 18). A partial view of the resulting 97-block grid is shown here in Fig. 1. The grid lines are smooth and nearly orthogonal, except near singular points formed by the block boundaries. This grid clearly demonstrates the flexibility of GRIDPRO/AZ3000 to generate CFD-quality grids for geometrically complex configurations, and in being able to concentrate grid points near solid surfaces. However, this software package does not lend itself easily to clustering grid points in the field away from the solid surfaces, e.g. along wake lines.

The computed pressure distributions for this configuration were compared with the experimental data by Vatsa et. al in reference 18. In general, the computed pressures agreed well with the measured data, and the resulting solutions indicated consistent treatment of the zonal interface boundary conditions. As expected, a large region of low velocity fluid was observed in the cove regions of the slat and the main airfoil, and on the upper surface of the flap. However, due to poor resolution in the wake regions, the computed velocity profiles did not correlate well with the experimental data (ref. 18).

### F/A-18 forebody/LEX

The next case that was considered for demonstrating the current multiblock code capability was the viscous flow over the F/A-18 forebody leading-edge-extension (LEX) geometry. The test conditions were chosen as  $M_\infty = 0.34$ ,  $Re_{\bar{c}} = 11.5 \times 10^6$ , and  $\alpha = 19^\circ$  to correspond to flight data (ref. 19). In the actual geometry, the LEX on the forebody merges with the wing leading edge. In the simpler model considered here, the grid at the end of

the LEX is extended downstream as a shroud of constant cross-section, which permits the application of a simple extrapolation condition at the downstream boundary. The effect of this simplification should be minimal on the flow over the forebody and the LEX. A similar approach has also been employed by Ghaffari et. al (refs. 20–21). A partial view of the 3–block grid used to model this configuration is shown in Fig. 2. A C-O type grid is used on the forebody (block 1), whereas H-O type grids are employed in the remaining blocks. The block boundaries are selected so that the configuration is subdivided into easily identifiable components.

The computational grid used in this study consists of approximately 750,000 mesh points. A 3.5 order-of-magnitude decrease in the residual of the continuity equation was obtained in about 325 work units (250 fine-grid iterations) in reference 10, which is considered quite good for such a fine mesh, and is significantly better than the convergence rate associated with non-multigrid type of codes. The global force coefficients converged in about half as many iterations. The computed surface-pressure compared favorably with the experimental data and indicated correct trends on both the forebody and the LEX except at the last axial station, where the effect of wing leading-edge missing in the computations becomes significant (ref. 10).

### Subsonic transport aircraft

The next test case that was considered in this study is that of a generic wing/body/engine/pylon configuration. The main reason for selecting this test case is to demonstrate the applicability of the current multiblock code for engine and airframe integration problems, typically encountered in advanced subsonic transport (AST) configurations. Current high-bypass-ratio engines have a very large frontal area and can have a significant effect on the flow field on the wing due to interference effects, which can alter the performance characteristics. Under transonic conditions, these interference effects cannot be predicted accurately with simple linear methods. The particular configuration considered here is a DLR transport aircraft with a high-bypass-ratio engine mounted on a pylon. An 11-block grid consisting of approximately 550,000 mesh points generated by Rossow and Ronzheimer (ref. 22) was used for computing the inviscid flow over this configuration. A partial view of this grid is shown in Fig. 3. The test conditions selected for these computations are  $M_\infty = 0.75$  and  $\alpha = 0.84^\circ$ , which are representative of cruise condition for this type of aircraft. The solutions for this case were reported by Vatsa, Sanetrik, and Parlette in reference 10. The computed surface-pressure contours were found to vary smoothly from one component to the next (ref. 10), which indicates a consistent and accurate treatment across block boundaries, since each component lies in a different block.

To assess the effect of the engine and pylon on the flow over the wing, the surface distributions on two cross sections that lie inboard and outboard of the pylon were compared with the computed pressures on the clean wing and body configuration in reference 10. Based on these comparisons, it was inferred that due to the interference effects caused by the engine and the pylon, the pressure peak flattens in the acceleration region on the lower surface, and the shock shifts forward on the upper surface of the wing, which results in reduced lift. The convergence history for this case was found to be very similar to the F/A-18 case, and a 3.5 order-of-magnitude decrease in the residual was achieved in 400 work units (ref. 10).

### Supersonic transport

Currently there is an enormous interest in United States and elsewhere for developing technology for the next generation of supersonic transports. Because of the high cost of testing a model at flight conditions, the designers must rely heavily on CFD analysis during the developmental phase of such vehicles. A 19-block,  $C^0$  continuous structured grid consisting of approximately one million grid points was created to represent a proposed configuration for demonstrating the current capability. A partial view of selected surfaces and zones for this configuration are shown in Figs. 4–6, to indicate the structure and topology of the computational grid. Each component was enclosed within a group of blocks, and the far field was filled with additional blocks. Singular points were introduced at block corners to facilitate an orderly matching of dissimilar topologies.

The Navier-Stokes solutions were obtained for a series of angles of attack at a cruise Mach number of 2.4. A typical run required 3–4 hours of cpu time on a single processor of the NAS Cray C-90 to obtain a converged solutions. The computed forces and moments have been compared with the experimental data, resulting in excellent agreement.

## STATUS OF EMERGING CAPABILITY

The flow solvers on structured grids are in a relatively mature stage of development at this time, and are undergoing mostly incremental changes. On the other hand, grid generation is a rapidly changing field that is experiencing evolutionary changes. In this section, we touch upon the activities in both of these disciplines that in our opinion will have significant impact on future CFD research.

### Grid generation

The biggest bottleneck in the grid-generation process occurs at the domain decomposition level, which is generally a labor-intensive interactive process. In the earlier stages of grid-generation software development, such an approach serves a useful purpose by giving complete control to the user regarding blocking strategies, thereby assisting in the assessment of relative advantages and disadvantages of different strategies. However, for routine grid-generation work, access to batch-oriented grid generation is preferable, in which the desired block structure can be selected via input files, and the grid generation can be completed in an hands-off manner. Two independent developments offer promise regarding domain decomposition. The ICEM/CFD has developed an object-based semi-automated hexahedral volume mesher for creating multiblock structured meshes, known as ICEM/HEXA. The user defines the initial structure or lets HEXA initialize the block structure around a given geometry. Input to the ICEM/HEXA can be either CAD geometry, NURBS surfaces or trimmed NURBS surfaces and NURBS curves. Mesh sizes can be defined on the family of CAD surfaces or individually on the block edges using edge meshing options. The grid is projected onto the underlying CAD geometry with minimum user interaction. The GRIDPRO/AZ3000 software package of Eisemann (ref. 5), on the other hand, employs a different strategy, in which the user specifies the block structure in a topological parametric space. The block boundaries in physical space evolve along with the solution to the grid coordinates. However, a few areas must be improved for this software to provide good quality CFD grids. First, the surface definition within AZ3000 needs a better representation than the bilinear patching that is currently implemented. A more flexible control of grid density is also required in predetermined regions in the field away from solid surfaces to allow clustering in high-gradient regions, such as wakes. Finally, a user-friendly input stream would greatly enhance the usability of this package.

A novel grid-generation methodology has been recently developed by a team of researchers from NASA Langley Research Center and the University of Leeds, England. In this method, known as Rapid Airplane Parametric Input Design (RAPID), a small set of design parameters and grid parameters govern the grid-generation process. The aircraft components (solid surfaces) are manifested through solution of a fourth-order partial-differential equation subject to Dirichlet and Neumann conditions. Volume grids are obtained through an application of the Control Point Method. This technique has been used to generate CFD-quality grids on airplane like configurations that consist of wing, fuselage, horizontal and vertical tails, and canards (ref. 23). This technique provides a medium level of fidelity in terms of surface representation; hence it is more suitable for preliminary conceptual design studies rather than final, detailed analyses.

Although truly automatic hands-off grid-generation capability for complete airplane configurations is still a dream, some recent developments could prove very helpful in parametric design studies. In many applications, CFD engineers are required to study the effect of small geometric changes on the aerodynamic performance of a configuration. The recently developed software package, known as the Coordinate and Sensitivity Calculator for Multidisciplinary Design Optimization (CSCMDO), can be used to generate volumetric grids which reflect small geometric changes in a configuration, given a baseline configuration and the grid associated with it (ref. 24).

This software is controlled via an ASCII user input file for execution in a batch environment. Once the grid on the baseline configuration is available from an independent source, CSCMDO provides the user with a simple, efficient tool for generating grids on perturbed configurations, that are encountered routinely in design optimization of specific class of aircraft.

### Flow solvers

Compared with grid-generation codes, the flow solvers for RANS are at a relatively mature stage in the development cycle. The state-of-the-art flow solvers can provide the steady-state solution to many problems of aerodynamic interest within a day on modern supercomputers. As discussed earlier, most of these flow codes can use  $C^0$  continuous grids, and many others can accommodate overlapped and patched grids. However, most of the solvers currently in use are non-conservative on patched boundaries and overlapped zones. Recently, a procedure has become available that can provide us with geometrically conservative interpolation coefficients for patched interfaces on multiblock grids (ref. 25). Work is in progress at NASA Langley Research Center to utilize these interpolation coefficients for developing conservative patched-grid flow solvers.

One of the major hurdles in delivering the latest flow-solver technology into the hands of industrial customers comes from lack of common standards for flow codes. The problem is more acute for multiblock structured codes (as opposed to single-block codes) due to the increased amount of zonal interface information required in such codes. For example, the time required to transfer a NASA-developed code in Boeing's project groups has been estimated by Boeing Engineers to be in years. In recognition of this difficulty, a NASA/Industry team consisting of participants from Boeing, McDonnell Douglas, NASA Ames, Langley and Lewis has been formed to alleviate this problem. The main objective of this team is to develop a Complex Geometry Navier-Stokes Analysis System (CGNS), which would standardize the input/output interfaces for major CFD codes under development at NASA and in the U.S. Aerospace industry. Currently, the team is finalizing the intellectual contents for the CGNS, and a prototype of the system is expected to become available before the end of this year. After the CGNS is developed and has been accepted by the CFD user community, the task of interchanging the flow solvers in a design system will become seamless and straightforward.

## CONCLUSIONS AND RECOMMENDATIONS

Block-structured-grid-based methods offer a viable choice for solving viscous flows over complex aerodynamic configurations. Boundary-fitted structured grids are well suited for resolving the thin viscous layers developing in the vicinity of solid surfaces at high Reynolds numbers typically encountered in flight. State-of-the-art structured flow solvers are known to be very efficient in computing aerodynamic flows in the presence of such embedded boundary layers.

The major stumbling block in routine application of structured-grid methodology to CFD applications is the grid-generation process. Progress is being made to make the surface modeling, domain decomposition, and volumetric grid-generation codes simpler to use. The block-boundaries that are kept fixed in physical space, can severely limit the overall smoothness and orthogonality of the resulting grids. Grid quality can be further enhanced by relaxing the requirement of  $C^0$  continuity across block interfaces, i.e. by permitting patched grids at the interfaces. It is much easier to control orthogonality and smoothness across blocks in patched grids. The effort and time required to generate such grids is less compared with equivalent  $C^0$  continuous grids.

The availability of different types of blocking strategies would be extremely helpful in the grid-generation packages of future. Both  $C^0$  continuous grids and patched grids should be available within the same package. Of course, the use of parametric surface definitions for the underlying geometry will retain the surface fidelity, and the resulting surfaces will correspond more closely with the parametric models from the original CAD definition. Another desirable feature is the ability to accommodate singular points at block boundaries for smoother merging of different topologies (e.g. H-H and C-O).



Future work should continue toward the development of batch-oriented grid-generation codes that require little or no human intervention, after the surface definition has been provided. Since domain decomposition can hinder the automation of block-structured grid generation, effort should focus on automating or simplifying the domain decomposition process. A knowledge-based system, which makes use of grid topologies commensurate with the geometric configurations will be extremely helpful in making this technology more attractive to CFD user community. In addition, grid adaptation strategies should be explored to make more efficient use of grid points.

## ACKNOWLEDGMENTS

The authors are deeply indebted to Dr. S.K. Agrawal of McDonnell Douglas Aircraft Co. for providing the surface grid for F/A-18 configuration and to Dr. C.C. Rossow of DLR (Germany) for providing the grid for subsonic aircraft used in this paper. The authors wish to acknowledge Mary Adams of NASA Langley Research Center for her invaluable help in creating the graphics for the supersonic transport configuration.

## REFERENCES

1. Slotnick, J.P., Kandula, 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, June 1994.
2. Weatherill, N.P.; Eiseman, P.R.; Hauser, J.; and Thompson, J.F. (eds.): Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Proceedings of the 4th International Conference, Swansea, Wales, Pineridge Press, 1994.
3. Arcill, A.S.; Hauser, J.; Eiseman, P.R.; and Thompson, J.F. (eds.): Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Proceedings of the 3rd International Conference, Barcelona, Spain, Elsevier Science Publishers B.V., 1991.
4. Smith, R.E. (ed.): Software Systems for Surface Modeling and Grid Generation, NASA CP-3143, Proceedings of a workshop held at NASA Langley, Hampton, VA, 1992.
5. Eiseman, P.: Multiblock Grid Generation with Automatic Zoning, Paper presented at the NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in CFD Solutions, 1995.
6. Rai, M.M.: An Implicit, Conservative, Zonal-Boundary Scheme for Euler Equation Calculations, Computers and Fluids, Vol. 14, No. 3, 1986, pp. 295-319.
7. Klopfer, G.H.; and Molvik, G.A.: Conservative Multizonal Interface Algorithm for the 3-D Navier-Stokes Equations, AIAA Paper 91-1601-CP, 1991.
8. Walters, R.W.; Thomas, J.L.; and Switzer, G.F.: Aspects and Applications of Patched Grid Calculations, AIAA Paper 86-1063, May 1986.
9. Volpe, G.: Conservation and Transparency at Internal Boundaries of Patched Grids, AIAA Paper 93-2930, July 1993.
10. Vatsa, V.N.; Sanetrik, M.D.; and Parlette, E.B.: Development of a Flexible and Efficient Multigrid-Based Multiblock Flow Solver, AIAA Paper 93-0677, Jan. 1993.
11. Vatsa, V.N. and Wedan, B.W.: Development of an Efficient Multigrid Code for 3-D Navier-Stokes Equations and its Application to a Grid-Refinement Study, Computers and Fluids, Vol. 18, No. 4, 1990, pp. 391-403.
12. Yu, N.J.; Allmaras, S.R.; and Moschetti, K.G.: Navier-Stokes Calculations for Attached and Separated Flows Using Different Turbulence Methods, AIAA Paper 91-1791, June 1991.
13. Agrawal, S.; Vatsa, V.N.; and Kinard, T.A.: Transonic Navier-Stokes Flow Computations over Wing-Fuselage Geometries, AIAA Paper 91-3205, Sept. 1991.
14. Marconi, F.; Siclari, M.; Carpenter, G.; and Chow, R.: Comparison of TLNS3D Computations with Test Data for a Transport Wing/Simple Body Configuration, AIAA Paper 94-2237, June 1994.
15. Milholen, W.E.; and Chokani, N.: Low Mach Number Performance of Three-Dimensional Compressible Navier-Stokes Codes, AIAA Paper 95-0767, Jan. 1995.

16. Turkel, E. and Vatsa, V.N.: Effect of Artificial Viscosity on Three Dimensional Flow Solutions, AIAA Journal, Vol. 32, No. 1, Jan. 1994, pp. 39–45.
17. Spalart, P.R.; and Allmaras, S.R.: A One-Equation Turbulence Model for Aerodynamic Flows, AIAA Paper 92–0439, Jan. 1992.
18. Vatsa, V.N.; Sanetrik, M.D.; Parlette, E.B.; Eiseman, P.; and Cheng, Z.: Multiblock Structured Grid Approach for Solving Flows over Complex Aerodynamic Configurations, AIAA Paper 94–0655, Jan. 1994.
19. Fisher, D.F.; Banks, D.W.; and Richwine, D.M.: F-18 High Alpha Research Vehicle Surface Pressures: Initial In-Flight Results and Correlation with Flow Visualization and Wind-Tunnel Data, NASA TM–101724, August 1990.
20. Ghaffari, F.; Luckring, J.M.; Thomas, J.L.; and Bates, B.L.: Navier-Stokes Solutions about the F/A-18 Forebody-Leading-Edge Extension Configuration, Journal of Aircraft, Vol. 27, Sept. 1990, pp. 737–748.
21. Ghaffari, F.: Navier-Stokes, Flight, and Wind Tunnel Flow Analysis for the F/A-18 Aircraft, NASA TP–3478, Dec. 1994.
22. Rossow, C.C. and Ronzheimer, A.: Multiblock Grid Generation Around Wing-Body-Engine-Pylon Configurations, Paper presented at the Third International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, Barcelona, Spain, June 1991.
23. Smith, R.E.; Bloor, M.G.I.; Wilson, M.J.; and Thomas, A.M.: Rapid Airplane Parametric Input Design (RAPID), AIAA Paper 95–1687, June 1995.
24. Jones, W.T. and Abolhassani, J.S.: A Grid Generation System for Multi-Disciplinary Design Optimization, Paper presented at the NASA Workshop on Surface Modeling, Grid Generation, and Related Issues in CFD Solutions, 1995.
25. Pao, S.P. and Hamid, A.: General Conservative Patch Algorithm for Multiblock Connectivity, NASA TP (under preparation).

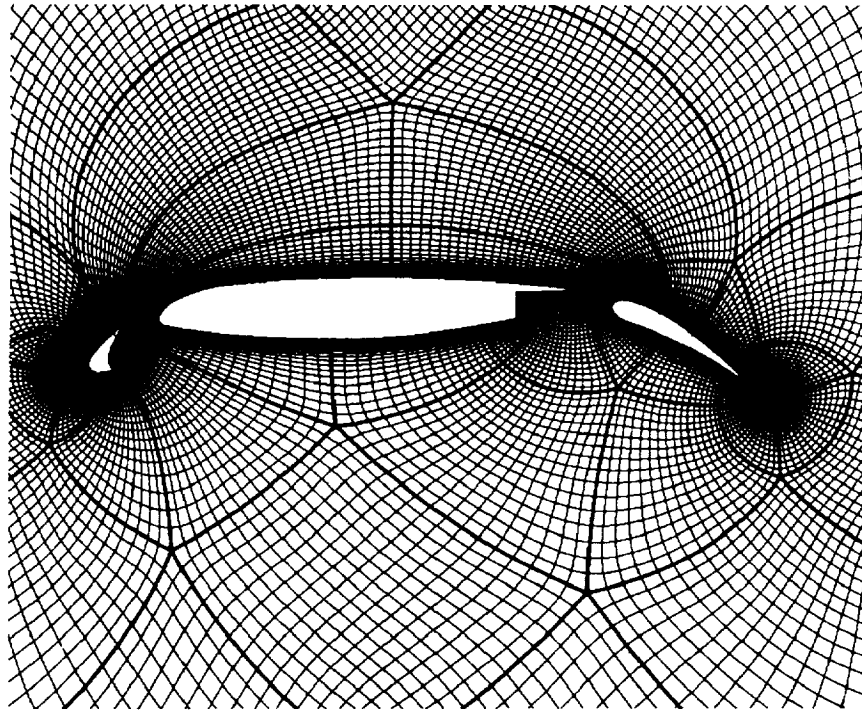


Figure 1.—Partial view of grid for 97-block multielement airfoil configuration.

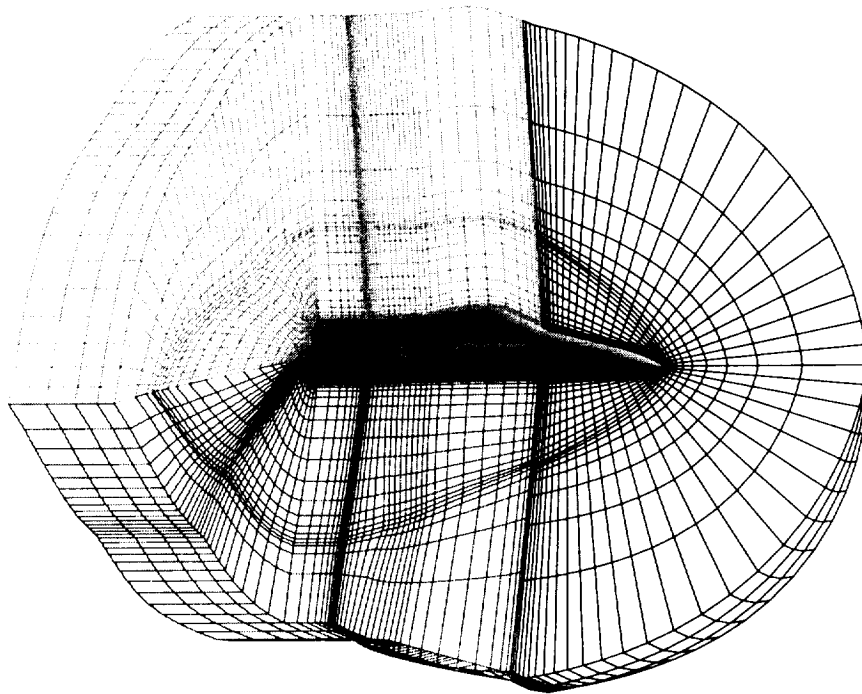


Figure 2.—Partial view of grid for 3-block F/A-18 forebody/LEX configuration.

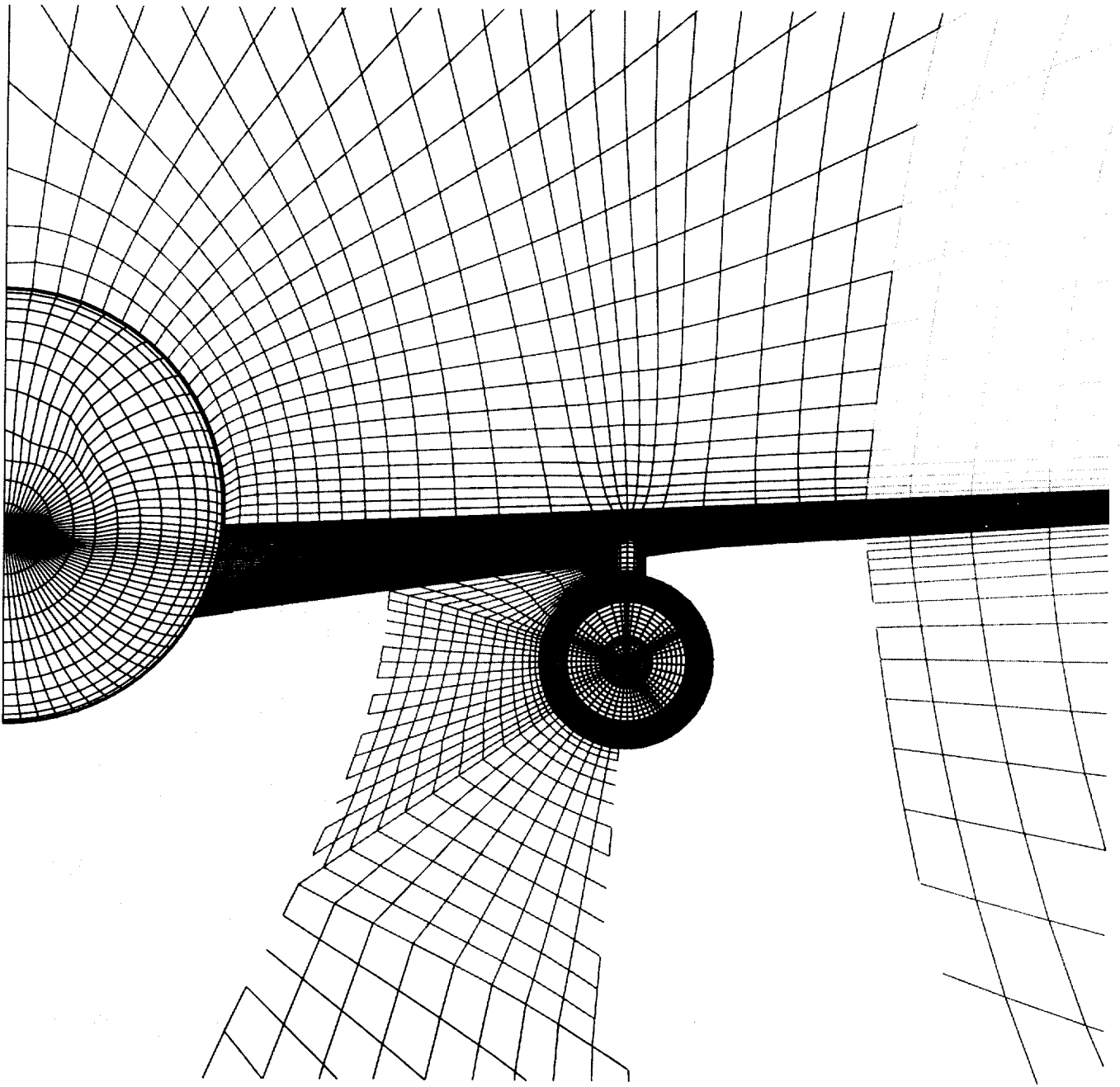


Figure 3.—Surface grid and streamwise cut for subsonic transport configuration.

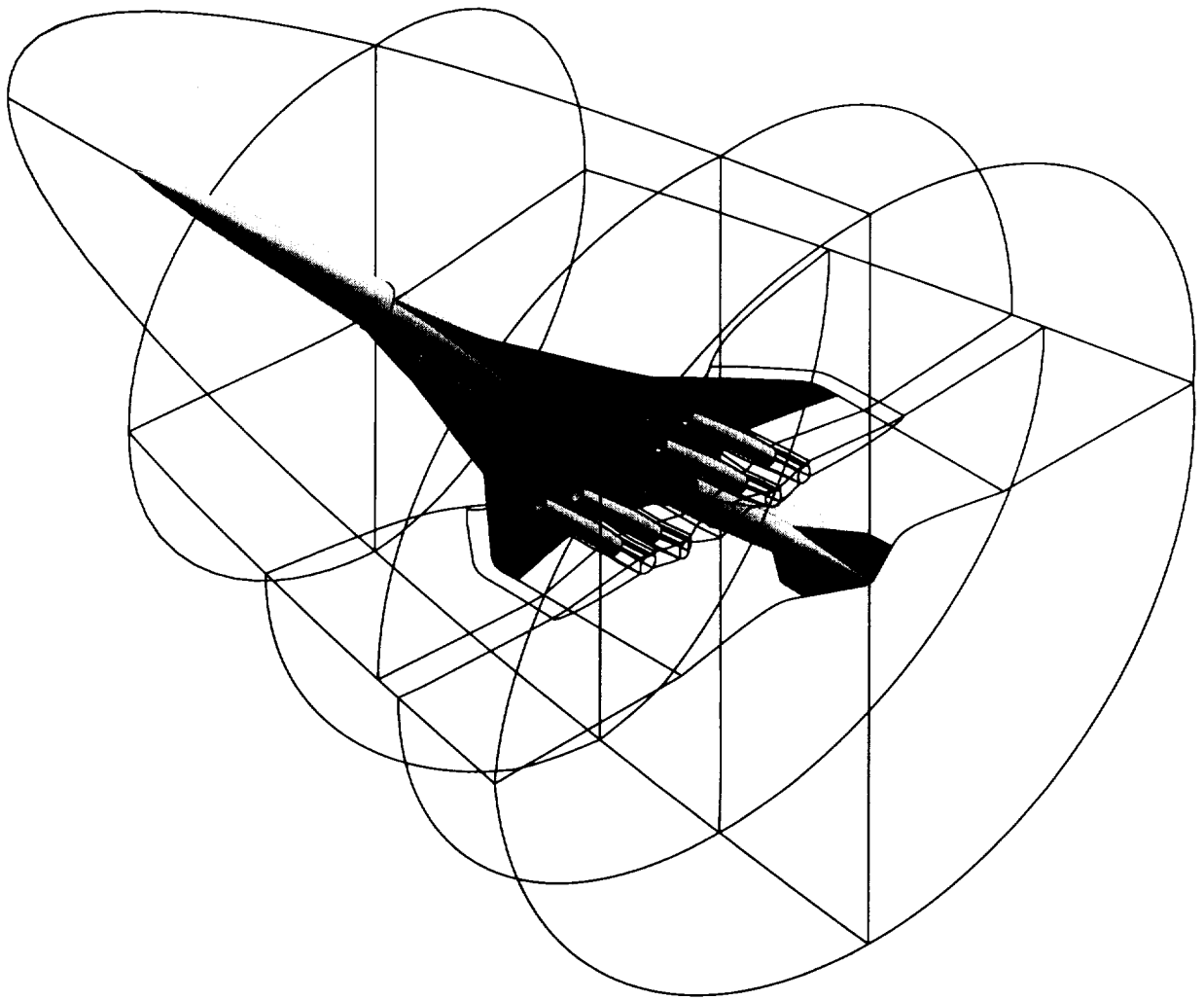


Figure 4.—Grid topology for 19-block supersonic transport configuration.

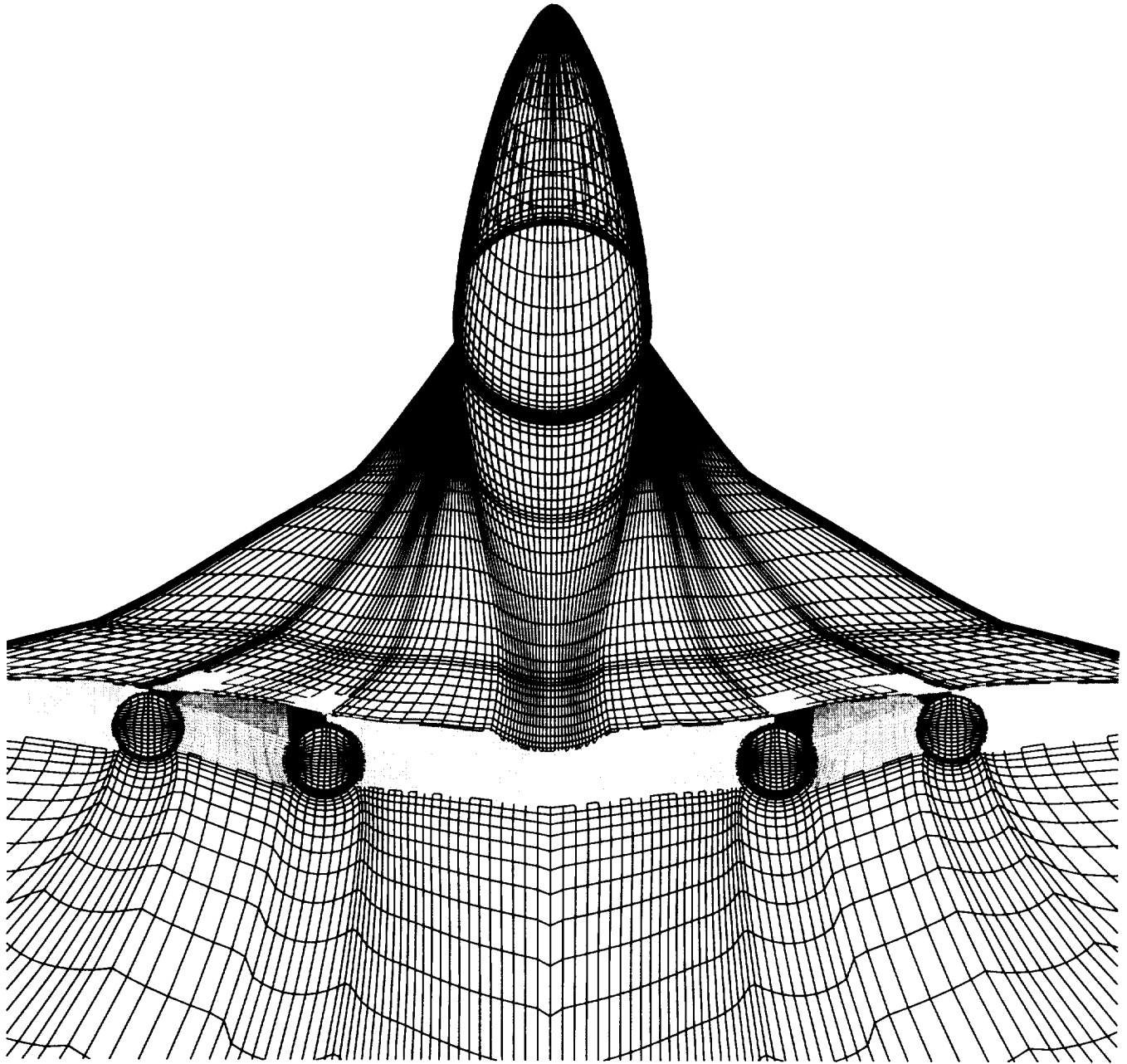


Figure 5.—Surface grid and streamwise cut for wing/fuselage/nacelle/diverter configuration.

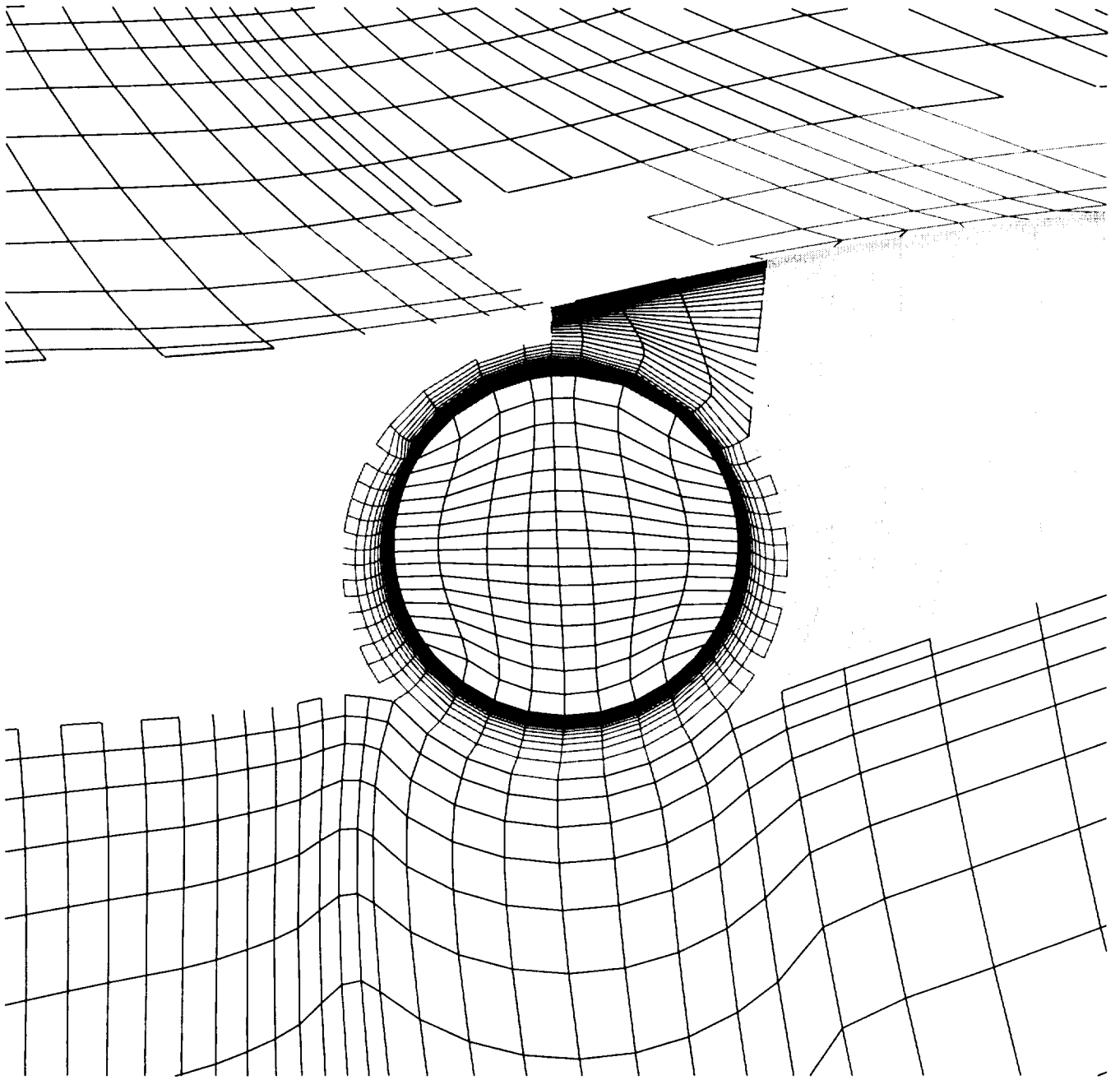


Figure 6.—Detail of nacelle/diverter topology.





STRUCTURED OVERSET  
GRID TECHNOLOGY

