1995122327

N95- 28748

# EVALUATION OF GRID GENERATION TECHNOLOGIES FROM AN APPLIED PERSPECTIVE

by

**Gary S. Hufford, Vincent J. Harrand, Bhavin C. Patel and Curtis R. Mitchell**
**CFD Research Corporation**
**Huntsville, Alabama 35805**

## ABSTRACT

An analysis of the grid generation process from the point of view of an applied CFD engineer is given. Issues addressed include geometric modeling, structured grid generation, unstructured grid generation, hybrid grid generation and use of virtual parts libraries in large parametric analysis projects. The analysis is geared towards comparing the effective turn around time for specific grid generation and CFD projects. The conclusion was made that a single grid generation methodology is not universally suited for all CFD applications due to both limitations in grid generation and flow solver technology. A new geometric modeling and grid generation tool, CFD-GEOM, is introduced to effectively integrate the geometric modeling process to the various grid generation methodologies including structured, unstructured, and hybrid procedures. The full integration of the geometric modeling and grid generation allows implementation of extremely efficient updating procedures, a necessary requirement for large parametric analysis projects. The concept of using virtual parts libraries in conjunction with hybrid grids for large parametric analysis projects is also introduced to improve the efficiency of the applied CFD engineer.

## 1. INTRODUCTION

Geometric modeling and Grid Generation are typically the most labor intensive tasks of a typical Computational Fluid Dynamics simulation. Factors which separate "good" tools from "bad" tools from the perspective of an applied CFD engineer include:

1. The ease/difficulty in obtaining a suitable surface definition for the particular grid generation scheme employed.

2. The ease/difficulty and the amount of time necessary to create an initial grid.

3. The ease/difficulty and the amount of time necessary to modify the initial grid both in terms of changing the number of grid points/distribution as well as the geometry, especially important for conducting parametric studies.

4. The amount of CPU required to run the Geometric Modeling/Grid Generation software as well as the amount of disk space required to store and re-create a model.

5. The applicability and efficiency of the CFD methodology for the particular grid created.

Each of these factors must be considered by the CFD engineer regardless of the type of grid generation/geometric modeling or CFD method employed.

In the commercial world the bottom line is the cost to complete a simulation. Cost manifests itself mainly in the form of man hours and computer equipment required. The overall time to complete a problem for a given methodology will always be related to the skill of the CFD engineer in using the available tools as well as the computer equipment available. Thus, evaluating and comparing methodologies can be quite subjective. Additionally "comparing methodologies" can often be confused with "comparing software tools." A poorly written software tool employing a given methodology tends not only to speak poorly of the software, but also of the methodology in general. This paper evaluates grid generation/geometric modeling technology in the areas of structured, unstructured and hybrid grids. The focus is on comparing the strengths and weaknesses of each methodology. Efficiency examples comparing methodologies are conducted using the current software tool, CFD-GEOM, developed at CFDRC which combines the geometric modeling and various grid generation methodologies in one integrated software package. The paper is divided into 8 additional sections: Geometric Modeling, Structured Grid Generation, Unstructured Grid Generation, Hybrid Grid Generation, Parts Libraries, Applied Comparison of Various Grid Generation Methodologies, CFD-GEOM: Current Status, and Conclusions.

## 2. GEOMETRIC MODELING

The definition of the surface for complicated geometric configurations is often one of the most laborious tasks in beginning the grid generation process. The surface definition is typically specified as a set of curves which interpolate to form a surface. For simple surfaces such as surfaces of revolution these interpolants are rather straightforward. However, for the generic three dimensional surface the problem is more difficult. For complicated surface definitions, most organizations use a CAD/CAM tool, which have the specific tools necessary for generating their geometries of interest.

The market is flooded with CAD tools used for design and manufacturing, however, the ability to transfer a CAD geometry to a surface geometry suitable for CFD grid generation is more limited. Two types of grid generation procedures are typical for CFD: structured and unstructured. The structured grid has been the historical mainstay of CFD while the unstructured grid is more common in structures applications. Specific strengths and weakness of each gridding procedure will be discussed later. However, the ability to transfer CAD geometries from any CAD/CAM system to a CFD grid generation system, either structured or unstructured, is crucial.

CAD tools can often be very specialized for specific applications and can be very large programs with hundreds of options. There is some attraction to incorporating a grid generation procedure into a CAD tool. However, in most CAD tools a significant amount of overhead makes this approach somewhat impractical due to memory requirements and CPU considerations. The closest tool to such a procedure is Icem-CFD[1], however, the incorporation into CAD is not direct. A transfer procedure from the CAD tool (DDN) to the grid generation tool (Mulcad or Hexa) is required. Other approaches use IGES[2] files as a starting point to create the grid. In this manner one can create his geometry with any CAD tool and import the geometry into the grid generation system. As long as the grid generation tool can support CAD transfer without information loss this is a viable approach, although two tools must be learned by the CFD engineer. The point of view held at CFDRC is that the grid generation tool should support basic CAD-like geometry creation functions, while allowing users to import geometries from any CAD system adhering to industry wide CAD file protocols (such as IGES). In this manner a CFD engineer does not necessarily need to learn a complicated CAD package to construct and grid simple geometries, however, if necessary he may. Additionally the grid generation tool should be able to manipulate or clean a geometry created using a CAD package to make it more suitable for grid generation purposes.

To properly facilitate CAD data transfer without information loss the grid generation system must understand the language of CAD. In modem CAD systems this requires using NURBS (Non-Uniform Rational B-Splines)[3]. NURBS have become a CAD industry standard for geometric definitions in CAD/CAM and Computer Graphics. NURBS have replaced the methods of Bezier[4], Coons[5], and Gordon[6] surfaces. NURBS have gained widespread popularity in the CAD community mainly for their ability to represent a wide range of surface types exactly. For example NURBS can express Bezier, Coons, and Gordon surfaces exactly as well as conic sections, quadratic surfaces and surfaces of revolution.

The CAD industry has also become heavily involved in Solid Modeling and Parametric Design principles. Solid Modeling[7], a technique developed in the late 1970's and early 1980's, contains information about the closure and connectivity of volumes of solid shapes. It is becoming an increasingly important part of computer aided design of solid physical objects for design, analysis, manufacturing, simulation and other applications. The advantage of a solid modeling technique over surface modeling is it's ability to form closed and bounded objects more closely related to physically realizable shapes. Solid models can also distinguish between the outside and inside of a volume, thus allowing mass property analysis to determine volume, center of gravity etc.

Parametric design is a design system which allows the user to select a predefined set of geometric constraints which can be applied to the geometry being created. Such a system can determine the positions of geometric elements specified by the predefined combinations of these geometric constraints. This system may also have an equation solver which allows a set of engineering equations to be used to set the values of dimensions based on either engineering parameters or values of other dimensions. In general the parametric design system allows the user to change specific parameters or constraints and quickly, if not automatically, obtain a variation from the previous design. This system is in stark contrast to the traditional CAD system which requires manual specification of each geometric entity within a model with no linking between the entities.

From the applied grid generation perspective a user must be able to take his CAD model and generate a grid. This requires data exchange from the CAD to grid generation system. This is identical to the need to transfer models across different CAD packages. The CAD industry has developed several standards of data transfer to support data exchange across CAD systems. The Initial Graphics Exchange Specification (IGES)[2] was the first such standard. IGES can handle a wide range of geometric primitives including NURBS. Initial IGES releases do not support parametric design or solid modeling information although these capabilities are planned in future releases. Other newer data exchange standards include PDES/STEP[8]. PDES (Product Data Exchange Specification) is an emerging standard for the exchange of product information among various manufacturing applications. The neutral exchange medium for PDES product models is the STEP (Standard for the Exchange of Product Model Data) physical file format.

The ability to begin with any geometric data (from any CAD source) and obtain a grid is a must for any grid generation system. When possible the parameters from the CAD package used in the design (ie. parametric design parameters, entity linking or solid modeling information) should be carried over from the CAD to grid generation system. The ability to perform this type of data transfer is only possible if industry wide standards are supported and adhered to by the CAD industry. As these standards develop they should be supported by the grid generation community. Currently NURBS are the highest level of surface grid generation supported by the standard CAD data transfer protocols and as such must be supported within the grid generation system.

## 3. STRUCTURED GRID GENERATION

The use of structured grids for CFD calculations is the historical mainstay of the CFD community. Viscous computations using structured grid methods are well established and the most advanced physical models are typically incorporated into structured codes. However, from the applied viewpoint, the grid structure often causes complications which are not physical in nature but are either geometric or related to the structure of the grid. This has led to unstructured grid CFD research where theoretically grid generation is easier. In practice the use of structured CFD grids, when used intelligently, can often be more efficient than using current unstructured techniques, especially when parametric analysis computations are the goal. This is possible since geometric information is inherently linked together in a structured grid through the grids structure. If small changes are made to the geometry (or grid distribution/density), the structured grid can be automatically and instantaneously updated if algebraic grid methods are used. In unstructured grids, these small changes require re-running of the unstructured grid generation procedure, a task which is trivial in terms of human labor but certainly not instantaneous. These types of issues will be the focus of the section entitled Applied Comparison of Various Grid Generation Methodologies later in this paper. In general a structured grid is defined as a grid which has a distinct i,j,k indexing. Four types of structured grid generation are typically used: *Algebraic, Elliptic, Hyperbolic and Advancing Layers/Advancing Normal.* Each of these techniques are most effective when object oriented multi-block grid generation is employed.

*Algebraic grid generation* is the most efficient means of grid generation in terms of computational speed and memory required. Algebraic grid methods rely on transfinite interpolation procedures to obtain a grid when the boundaries of the grid have been specified with a user-described distribution. The algebraic functions then interpolate to obtain a surface or volume grid. Many different interpolant procedures have been used to obtain a desired grid. These include the standard transfinite interpolant[9], Ericksson[10], Hermite orthogonal[11] among others with hybrid approaches blending the best features of each method. In all cases these methods are extremely fast (essentially instantaneous even on the smallest current workstations) and require very little memory. As such very fine grids can be created very quickly. Typically these methods are used in a multi-block patched procedure. In this manner the CFD engineer de-composes the entire CFD domain of interest into patches (or blocks) and uses the algebraic methods for each block. He can then combine these blocks or keep them separate depending on the application. This method of grid generation requires more human effort than other methods to obtain an initial grid, however, the task can be made much easier with graphical CAD-like software tools. Once an initial grid is constructed, small modifications can be made and a new grid created instantaneously due to the geometric linking. The ability to instantaneously update a structured grid with minor modifications is dependent on the software design, however, it is possible and will be demonstrated in Section 7.

*Elliptic grid generation* was developed to "automatically" obtain a smooth internal grid distribution. Elliptic methods[12,13] rely on the solution of elliptic partial differential equations to obtain a smooth internal grid distribution. The user may specify controlling parameters which effect the grid clustering and grid orthogonality both internal and at the boundaries. Typically these methods work within a single patch or block. Thus, a user must still define the grid topology as with the algebraic methods, only now the solution of elliptic partial differential equations replaces the algebraic functions. The main advantage of this method over algebraic methods is it can provide for a smoother grid. However, in practice the user must play with the controlling parameters to obtain the grid he desires. Additionally elliptic techniques are much more computationally expensive and require much more memory usage than algebraic techniques. Thus, very fine grids can take a long period of time to create and require a relatively large amount of computer memory when compared to algebraic methods. In general the elliptic methods should only be used over an algebraic method when the algebraic functions are not providing the quality of smoothness desired or needed for the CFD solver. In practice this is a very small percentage of the time.

*Hyperbolic grid generation* is a grid generation technique based upon the solution of hyperbolic differential equations. Hyperbolic methods[14] were introduced to automatically generate a body-fitted orthogonal grid for external flow applications. The method uses an initial surface point distribution along with cell volume and orthogonality constraints to solve a set of hyperbolic differential equations to obtain the grid. In general the hyperbolic methods are one to two orders of magnitude savings in computer time over elliptic techniques[15] however, they are generally less robust than the elliptic techniques. They are also more restricted in the types of problems they can address since the outer boundary of the mesh cannot be specified. Currently the hyperbolic technique is generally used for Chimera applications[16] where the grid systems of several objects are allowed to overlap. For the hybrid grid generation techniques discussed below the hyperbolic system may be useful if a control is provided to prevent grid overlapping of several objects. This is analogous to setting a normal distance control. This requirement is the opposite of the Chimera philosophy to allow or enforce overset grids. In the hybrid system the gaps would then be filled with unstructured grid.

*Advancing Layer* or semi-structured grid generation[17,18] originated in the unstructured grid generation community. It is used to obtain viscous boundary layer grids near boundaries. Essentially the method "pops" a grid from a curve or surface in an automatic fashion based on the normal vectors emanating from the surface grid. The method does this in a layered fashion building from the surface. If a structured quadrilateral surface grid is used the resulting grid is essentially a structured hexagonal grid near the surface, as such it is labelled in this paper as a structured grid method. The surface grid could also be an unstructured triangular grid resulting in a prismatic boundary layer grid that has a structure in the normal direction. Often times the near wall hexagonal grid is sub-divided to form tetrahedrons or the number of layers emanating from each normal vector is unequal resulting in an artificially generated unstructured grid. The true unstructured portion of the method results when this inner layer of semi-structured grid is merged with other unstructured grid methods in an "automatic" fashion. When quadrilateral surface grids are used this method is ideal for obtaining structured boundary layer grids very quickly. The debate about whether one should use prisms or hexahedrons in the boundary layer hinges on two questions. First, is it easier to obtain a controllable surface grid using a quadrilateral or triangular method, and second which grid type is more accurate and efficient in terms of the CFD solver. The use of NURBS for surface grid generation and its tensor product definition leaves the first question open while the second question is still being debated/reviewed in the flow solver community.

## 4. UNSTRUCTURED GRID GENERATION

A grid is said to be unstructured if the cells comprising the grid are not in any particular i,j,k order. Unstructured grids typically consist of triangles in 2-D and tetrahedrons in 3-D, although this does not have to be the case. Recently, unstructured discretization procedures have been extended to include prisms, hexahedra as well as combinations of these elements.

Unstructured grid generation has been dominated by three methods: Delaunay or Voronoi-based[19,20] techniques, modified quadtree/octree[21], and the Advancing Front[22] methods. Each of these approaches have been successfully applied for a variety of complex configurations. Delaunay methods require an initial distribution of nodes throughout the domain which are connected to form the unstructured grid. The points are connected to satisfy the Delaunay criteria to *promote* the generation of isotropic (or equi-angle) elements. In 2-D the elements are triangles while in 3-D the elements are tetrahedrons. By the very nature of the Delaunay criteria, the controlled creation of stretched cells in boundary layer regions is problematic. The Delaunay method also has the requirement that the points must be initially placed before they can be connected to form a grid. The modified quadtree/octree approach is another commonly used procedure which is based on subdivision algorithms to generate unstructured grids.

The Advancing Front Method is a method to both place points and create cells simultaneously. In the advancing front method the type of cells created can either by isotropic or stretched. The type of cells desired is specified on a background grid which encompasses the entire domain. Typically a set of Poisson equations is solved on the background grid[23] to control the cell size, cell stretching and cell stretching direction. In 2-D it has been shown that high fidelity stretched unstructured grids can be created with up to 20:1 base to height aspect ratios. In 3-D uni-directional stretching has been achieved.

The use of either Delaunay, octree or Advancing Front methods can obtain fairly smooth isotropic unstructured grids. Some general stretching can be achieved using Advancing Front methodology, however, for extremely stretched cells necessary to resolve near wall effects and shear layers (1000:1 base to height ratio is typical) semi-structured[18] or layer[17] generation methods are used. The Advancing Layer/Advancing Normal method is typically exercised to refine regions of the domain which require highly stretched mesh features. The grid generation process then proceeds using either the advancing front or Delaunay method. This results in a completely unstructured grid with semi-structure mesh characteristics near boundaries. In general the unstructured grid generation methods are more computationally intensive than the structured grid generation techniques, especially as the cell numbers become large.

Theoretically the unstructured grid generation techniques require less manual labor than the structured techniques at a cost of CPU efficiency, i.e. algebraic vs. subdivision, subdomain removal, Voronoi-based. For parametric design purposes where many grids/geometries must be re-created over a period of time this trade-off may or may not be acceptable depending on the degree of the design change.

## 5. HYBRID GRID GENERATION

The term hybrid grid is defined in this paper as a grid which contains different grid types in different regions of a flow field domain. Generally the hybrid grid will contain a quadrilateral/hexahedral or prismatic grid in near wall boundary layer regions and a triangular/tetrahedral or adaptive cartesian grid[24] in inviscid regions. The appeal of hybrid grids is that more accurate and advanced structured flow solver methodologies can be used in viscous flow regions while the unstructured techniques can provide more flexibility in the gridding process. From the point of view of the applied CFD engineer, the hybrid grid gives much greater grid creation flexibility. When coupled to an object oriented grid generation software tool the CFD engineer can use different grid generation procedures in different flow regions utilizing the strengths of the different procedures and eliminating the weaknesses. The major drawback to using hybrid grids is that the CFD code must become more general, either handling structured and unstructured domains differently or handling multiple element grids in a purely unstructured fashion.

A significant advantage in using hybrid grids is that the grid can be separated into parts or objects. This is analogous to the multi-domain structured approach only now any grid type may be used in a particular domain. Since the flow field region is decomposed the individual domains can be modified as long as the interfaces remain unaffected. The use of hybrid grids is a relatively new development in the CFD world. It has enormous potential to speed up the turn around time for parametric analysis CFD computations when coupled to the concept of a parts library. The use of hybrid grids in conjunction with the parts library will be demonstrated in the next section.

## 6. PARTS LIBRARIES

An extremely important concept in the CAD design community is the use of Parts libraries. Typically, a CAD engineer will comprise individual parts of a design and piece them together to form the complete design. The concept of extending the parts library to include the grids is very attractive for many CFD analysis projects. This allows the CFD engineer to construct a grid around an object, save the geometry and grid together in a library, and then piece the parts together in the final analysis project. This type of system is very suitable for hybrid grid generation. The near wall boundary layer grid can be stored with the geometry in a library, pieced together and then a general unstructured grid generation algorithm can be used to fill in the gaps. The Parts library concept is also extremely suitable in the Chimera method where the grids of each part are independent of one another, and can also be used in the patched structured grid method as well. The use of the parts library in the grid generation process is extremely suitable for parametric analysis.

The concepts of hybrid grid generation in conjunction with the use of Parts libraries for grid generation is very new. Therefore, an example of the power of these concepts will be demonstrated using a 3-element airfoil example. Figure 1 shows the base parts of the 3-element airfoil. Figure 2 shows two scaled versions of the slat and flap parts. Each of the individual parts is surrounded by a structured grid, Figure 3, and saved in a library. The outer boundaries for each of the slat and flap configurations, Figure 4, are identical so that they can be interchanged. Figure 5 shows the unstructured grid that fills in the gaps for the baseline configuration. An Advancing Front Technique with source controls[23] is used to obtain interface matching along the boundaries. Now if any of the other slat or flap configurations is desired they can be automatically interchanged without re-running the unstructured grid domain as shown in Figure 6. Additionally, for near wall O-grids, like that of the flap, if one wishes to refine the structured grid in the normal direction one can without any re-gridding of the unstructured part. In three dimensions where pure unstructured grid generators require a fair amount of CPU, this use of the hybrid grid generation concept along with the parts library concept saves a considerable amount of time for large parametric analysis projects. Additionally if the slat or flap angles of attack are changed or if the geometry is such that the outer boundaries cannot be efficiently made identical, only the re-solution of the unstructured domain is necessary. In most cases no new user input, only the geometrical change, is necessary when re-computing the unstructured domain. Figure 7 shows the grid resulting when the slat is pitched upward at 20 degrees.

## 7. APPLIED COMPARISON OF VARIOUS GRID GENERATION METHODOLOGIES

The grid generation methodology to use for a particular application will often be different depending on the goals of the project. Often times CFD engineers are limited by the type of solver they are using and by the types of grid generation tools available. The approach favored at CFDRC is to use Hybrid grid generation along with Parts Libraries to maximum the efficiency of the applied CFD engineer, especially for large parametric analysis projects. Each particular grid generation methodology has particular strengths and weaknesses. This section will look at the various strengths and weaknesses, with appropriate examples, of the following grid generation techniques: Algebraic structured multi-block, Elliptic, Unstructured Advancing Front, Advancing Layer/Advancing Front Unstructured, and Hybrid. When possible specific data (based on CFDRC experience) such as CPU time and human effort statistics will be given for each of the examples provided. As is always the case with human effort statistics they are very dependent on the user's experience level with a given methodology, and as such are somewhat subjective. The use of these statistics should not be taken as absolute but in relative comparison to one another.

### Algebraic Structured Multi-Block

The strength of the algebraic multi-block methodology is its CPU speed and limited memory requirements. Very fine grids can be created very quickly. Additionally the ability to make small modifications and obtain instantaneous grid updates is possible (and demonstrated using CFD-GEOM) due to the linking between the geometry and the grid. The weaknesses of the algebraic multi-block methodology are the increased human effort to obtain an initial grid (ie. setting up the grid topology) and the possibility that the grid structure wastes cells. The example chosen for demonstration purposes is a grid of the B-1A Escape Capsule.

Figure 8 shows the multi-block structure of half of the B-1A capsule grid and the surface grid. There are 22 grid blocks to control the grid clustering which are combined to form a single composite block with blockages. The definition of the geometry was obtained from a Plot3D surface point set and broken up into the various topological domains in approximately 4 hours using CFD-GEOM. This includes creating additional geometry to form a topology and the topological refinement process to easily refine the grid at a later date. The grid consists of 90x75x41 grid lines comprising 263,440 cells. The final algebraic grid at the centerline is shown in Figure 9. Once the initial set-up is complete small modifications to the geometry or grid distribution are essentially instantaneous. For example using the CFD-GEOM software package the front spoiler can be rotated (Figure 10) to any angle and the corresponding geometry and topology which are effected is automatically updated in seconds (using an SGI Indy with R4600 processor). Although the initial time to set-up the topological structure is sometimes intimidating, an experienced user can use the advantages of the topology to easily and quickly modify the grid with instantaneous updating.

### Elliptic

The strength of the elliptic method is the ability to obtain a smooth internal grid "automatically," reducing the need of the CFD engineer to pay quite so much attention to the grid topology. The weaknesses are that an initial grid is needed to begin the process (usually an algebraic grid), the CFD engineer must play with some clustering parameters to obtain a suitable grid, and the solution of the elliptic differential equations is CPU intensive and dependent on the grid size. An elliptic method has been applied to the algebraic grid of the B-1A Escape Capsule for comparative purposes.

Figure 11 shows the resulting elliptic grid obtained for the escape capsule. To converge the final elliptic result (using an in-house CFDRC elliptic grid solver using the methodology of Shieh[13]) takes approximately an hour on an IBM R6000 workstation. To obtain the proper clustering functions before obtaining the final result also takes some time. In general it takes 3 to 4 tries to find the proper clustering mechanism. In certain instances (for example at the capsule wing tip) the clustering functions and surface distributions need to be locally refined which takes some more time. For this particular case it took approximately 2 hours to obtain the proper clustering. Additional time is necessary to obtain the initial guess. In this case the final algebraic grid in Figure 9 was used, although if an elliptic method was desired from the start, the set-up time could probably have been reduced in half. The biggest disadvantage of this method is that to make a small geometry change requires re-computation of the elliptic equations (another hour). If the grid is increased in size these computational requirements increase. If the size is increased by orders of magnitude computer memory requirements can also become an issue. Thus, this method becomes inefficient if a parametric analysis or grid refinement study is required.

### Unstructured: Advancing Front

The strength of all unstructured techniques is that the applied CFD engineer does not have to set-up a grid topology, thus it is less user intensive. Once the geometry is defined the engineer specifies some controlling parameters and engages the unstructured grid method. The weaknesses of the unstructured method are the difficulties in controlling the resulting grid in both cell size and cell stretching primarily due to the difficulty in creating adequate background grids and developing controlling mechanisms for complex highly-curved geometries, and the high CPU and memory requirements for the algorithm. Progress has been made in 2-D cases for generic grid control in both cell size and stretching. Recent results show that stretching up to 20:1 aspect ratio[23] can be robustly achieved for arbitrarily complex geometries in the advancing front algorithm, however this is not nearly enough in boundary layers. In 3-D uni-directional stretching control has been achieved and generalized stretching for complex bodies is possible using the methods of Reference 23, however, sufficient stretching in boundary layers will be limited as in the 2-D case.

For comparative purposes an unstructured Euler type grid has been constructed for the B-1A escape capsule. The structured grids created for the capsule are somewhat stretched near the surface in the normal direction. For the unstructured grid this was not attempted. The resulting unstructured grid, shown in Figure 12, contains ~245k cells and took ~3.5 hours to complete on an SGI R4400 computer. Note that the grid is suitable only for Euler computations. With CFD-GEOM, the set-up time to define the capsule surfaces beginning with the Plot3D surface grid took approximately 5 minutes while the iterations to obtain the proper clustering took approximately 15 minutes. The strength of the method is the minimal set-up time, however, the CPU to obtain the grid is considerable. Additionally as the grid is refined the CPU requirement increases significantly and if the grid is refined by orders of magnitude the computer memory requirements become an issue.

### Unstructured: Advancing Layer/Advancing Front

To combat the inability of the Delaunay or Advancing Front Methods to create a suitably stretched viscous grid, several researchers[17,18,25] have used an Advancing Layer or Advancing Normal approach to create semi-structured grids near boundaries and then merged this in an "automatic" way to an Advancing Front methodology when stretching requirements are reduced. The type of grid typically produced using this method is shown in Figure 13. Sometimes the quadrilateral near boundary elements are sub-divided to obtain a purely triangular grid.

The strength of this method, as in the pure Unstructured methodology, is the limited amount of user set-up. However, the computational requirements are still significant. In Advancing Front regions they are similar to the CPU requirements quoted above. In Advancing Layer regions the CPU requirement is less extensive, although the memory requirements are similar to the Advancing

Front requirements (much larger than for a structured grid) due to the indirect indexing of the created cells. The biggest weakness of such an approach is if small changes are made to the geometry or grid refinement is required the algorithm must be re-engaged, a significant computational expense if a parametric analysis is required.

Hybrid: Structured Algebraic/Unstructured Advancing Front

The use of purely Unstructured methodologies, such as the Advancing Front method, reduces greatly the amount of work a CFD engineer must do to obtain an initial grid. However, the speed and limited memory requirements with which algebraic (including Advancing Layer/Normal techniques) structured grids require and the speed with which the grids can be updated is extremely attractive for parametric analysis. Thus, the hybrid grid approach can take advantage of the best of the structured and unstructured methodologies. Additionally the hybrid grid method is ideally suited for the Parts Library concept described in Section 6. The near wall structured grids can be saved in a library with the geometry and pieced together. Then the automated unstructured methodology is used to fill in the gaps. A good example of this type of technique is given in Section 6, which demonstrates the advantages of the Hybrid grid methodology along with the Parts library concept.

## 8. CFD-GEOM: CURRENT STATUS

CFD-GEOM is an interactive geometric modeling and grid generation software package with fully integrated multi-block structured, unstructured and hybrid grid generation capabilities. CFD-GEOM is targeted towards the CFD end-user; the main goal is to enable a non-expert user (in geometric modeling and grid generation) to interactively create a moderately complex model relatively easy. CFD-GEOM uses the Motif graphical user interface environment in conjunction with the IRIS/GL and OPEN/GL graphics libraries to support compatibility across many different workstation platforms.

The **philosophy** of CFD-GEOM is that a single grid generation methodology is not suitable for all applications, as such different grid generation methodologies must be supported within the same software environment. In addition the geometric modeling capabilities are entirely based on the NURBS primitive allowing a CAD environment to develop geometry within the grid generation software. The focus of the overall CFD-GEOM software design is to **fully integrate** geometric modeling and various grid generation technologies into one data base. Thus, all geometric data, structured grid data, and unstructured grid data share a common data base and as such are completely linked together. The linking facilitates **automatic database updating** when any piece of the database is changed. These changes can include geometry changes or grid (density/distribution) changes. For example if a geometric entities shape is changed, immediately the effect of this change, and the number of other entities which are affected, is noticed throughout the rest of the database. Since the scope of the change in the database is known, only those parts of the database changed are updated. This facilitates the ability to update the minimum amount of information necessary. This communication throughout CFD-GEOM facilitates the instantaneous updating demonstrated in Figure 10. Additionally this software design allows for easy implementation of new technologies into the pre-designed data base.

The CFD-GEOM environment currently supports NURBS based geometric modeling capabilities, algebraic multi-block structured grid generation, multi-block unstructured advancing front grid generation, and hybrid multi-block structured/unstructured grid generation. Each of these functional operations is incorporated into a common database allowing changes of any kind to be felt immediately throughout the database. The functionality of each operation resembles separate software packages although they are fully integrated. For example, the geometric modeling capability of CFD-GEOM can be used as a simple CAD package irrespective of the grid generation capabilities and the grid generation capabilities, both structured and unstructured, can be used without the geometric modeling capability through IGES. Additionally a set of CFD-GEOM models can be stored into a virtual parts library and subsequently merged at will. This allows a user to create separate geometries and grids and mix and match them to form a new model. CFD-GEOM also filters all entity duplication between merged models, during IGES or Plot3D reading and at creation.

Specific geometric modeling capabilities within CFD-GEOM include: NURBS based toolkit to interactively define points, (poly) lines, arcs, free-from curves and surfaces; NURBS curve creation using arcs, splines, point-tangent, point revolution and point extrusion; NURBS surface creation using ruled, extruded and revolved surfaces; Interactive NURBS curve and surface modification by interactively modifying the control points and weights; Automatic intersection of curve and line sets including multiple intersections; Automatic intersection of surface sets up to all available surfaces simultaneously including multiple folded surfaces.

Specific topological and structured grid generation capabilities within CFD-GEOM include: Edge creation composed of single or multiple curves; Face creation from a set of 4 edge sets or directly on a NURBS surface; Block creation consisting of single or multiple faces sets; Block compositing to form a single composite block from a block set; Topology editing to re-orient a single block or to re-orient an entire connected block structure based on a single blocks orientation; Automatic grid updating when mesh density/distribution is changed; Edge Linking by point number, point distribution or both allowing the user to specify how changes propagate through the grid; Boundary condition identifiers to allow the flow solver to identify the location and type of boundary condition for a specified entity; Interactive definition/alteration of mesh point distribution; Multi-to-one connectivity for 2-D and 3-D blocks.

Specific unstructured and hybrid grid generation capabilities include: Automatic 2D/3D advancing front mesh generation; Source controls including point, line, curve, area and surface sources; automatic or manually controlled stretched cartesian background grids for grid control; General grid smoothing; Interfacing to structured mesh blocks for hybrid mesh generation.

Additionally certain operations can be completed on any entity including translation and rotation with automatic database updating, duplication and translation, and duplication and rotation. These functions in combination with the general curve and surface editor allow the user to make changes in the geometry (or add to the geometry) with immediate database updating, a capability specifically designed for large parametric analysis projects requiring a large number of small design changes.

Figure 14 shows a multiple, differing diameter pipe intersection created within the CFD-

GEOM environment. A key feature of CFD-GEOM is the ability to arbitrarily organize geometry and/or grid entities into parts. Each part can be arbitrarily manipulated and placed within any CFD-GEOM model. For example, the pipe section tear-off, shown in the lower left hand corner of Figure 14, can be rotated 180 degrees and glued back to the main model as shown. Other global part operations include translation, scaling, rotation and duplication. Additionally arbitrary shape modifications can be made by manipulating NURBS control points or ordered modifications such as face extrusion or rotation can be completed.

## 9. CONCLUSIONS

Based on comparisons between the various structured and unstructured grid generation methodologies it is apparent that one methodology is not universally suited for all problems. Couple this conclusion with the fact that a particular flow solver capability may only be available for a certain type of grid (ie. structured grid flow solvers are typically richer in physical models than unstructured flow solvers) and it is apparent that a single grid generation methodology is currently insufficient for CFD applications. Additionally, it is absolutely essential that the grid generation process incorporate advanced CAD geometric modeling capabilities to aid the CFD engineer in large parametric analysis projects. These realizations have led to the development of the CFD-GEOM geometric modeling and grid generation software system. The design of CFD-GEOM incorporates full integration between geometric modeling, structured grid generation, unstructured grid generation, and hybrid grid generation which allows for automatic and efficient database updating and the development of parts libraries. The software is specifically designed to aid the applied CFD engineer in large parametric CFD analysis projects.

## ACKNOWLEDGEMENTS

This work has been funded by the IR&D program of CFD Research Corporation. The authors would like to acknowledge Mr. Sami Habchi and Dr. Simon Ho for their help in generating the elliptic B-1A escape capsule grids, Drs. Ashok Singhal and Andrzej Przekwas for their support of this work, and Ms. J. Swann for preparation of the manuscript.

## 10. REFERENCES

1. Bertin, D., Lordon, J. and Marcaux, V., "A New Automatic Grid Generation Environment for CFD Applications," Aerospatiale, 1992.
2. Reed, K., Harrod, Jr., D. and Conroy, W., "The Initial Graphics Exchange Specification (IGES) Version 5.0," NISTIR 4412, NIST, Gaithersburg, MD, 1990.
3. Piegl, L and Tiller, W., "Curve and Surface Constructions Using Rational B-Splines," Comput. Aided Des., vol. 19, pp. 485-498, 1987.
4. Farin, G., Curves and Surfaces for Computer Aided Geometric Design, New York: Academic Press, 1988.
5. Coons, S., "Surfaces for Computer Aided Design," Technical Report MIT, 1964.
6. Gordon, W., "Free-Form Surface Interpolation through Curve Networks," Technical Report GMR-921, General Motors Research Laboratories, 1969.

7. Mantyla, M., An Introduction to Solid Modeling, Rockville, MD: Computer Science Press, 1988.

8. Wilson, P. and Kennicot, P., "ISO STEP Baseline Requirements Document (IPIM)," ISO TC184/SC4/WG1, Document N284, 1988.

9. Gordon, W.J., "Blending-Function Methods of Bi-Variate and Multi-Variate Interpolation and Approximation," SIAM Journal of Numerical Analysis, vol. 8, no. 1, pp. 158-177, March 1971.

10. Eriksson, L.E., "Generation of Boundary-Conforming Grids Around Wing-Body Configurations Using Transfinite Interpolation," AIAA Journal, vol. 20, no. 10, Oct. 1982.

11. Chawner, J.R. and Anderson, D.A., "Development of an Algebriac Grid Generation Method with Orthogonal and Clustering Control," Numerical Grid Generation in Computational Fluid Dynamics and Related Fields, A.S. Arcilla, J. Hauser, P.R. Eiseman, J.F. Thompson (editors), Eselvier Science Publishers B.V. (North-Holland), pp. 107-117, 1991.

12. Shieh, C.F., "Three-Dimensional Grid Generation Using Elliptic Equations with Direct Grid Distribution Control," AIAA-83-0448, AIAA 21st Aerospace Sciences Meeting, Reno, NV, Jan. 1983.

13. Thomas, P.D. and Middlecoff, "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA Journal, pp. 652-656, June 1980.

14. Steger, J.L. and Chaussee, D.S., "Generation of Body-Fitted Coordinates Using Hyperbolic Differential Equations," SIAM Journal Sci. Stat. Comput., vol. 1, no. 4, Dec. 1980.

15. Chan, W.M. and Steger, J.L., "A Generalized Scheme for Three-Dimensional Hyperbolic Grid Generation," AIAA-91-1588-CP, presented at 10th AIAA CFD Conference, June 1991.

16. Steger, D.L., Dougherty, F.C. and Benek, J.A., "A Chimera Grid Scheme," in Advances in Grid Generation, K.N. Ghia and U. Ghia, eds. ASME FED-5, pp. 59-69, 1983.

17. Pirzadeh, S., "Unstructured Viscous Grid Generation by Advancing Layers Method," AIAA-93-3453, AIAA 11th Applied Aerodynamics Conference, Aug. 1993.

18. Lohner, R., "Matching Semi-Structured and Unstructured Grids for Navier-Stokes Calculations," AIAA-93-3348-CP, 1993.

19. Meshkat, S., Ruppert, J. and Hungwen, L., "Three-Dimensional Grid Generation Based on Delaunay Tetrahedrization," Numerical Grid Generation in Computational FluidDynamics and Related Fields, A.S. Arcilla, J. Hauser, P.R. Eiseman, J.F. Thompson (editors), Eselvier Science Publishers B.V. (North-Holland), pp. 841-853, 1991.

20. Watson, D.F., "Computing the N-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes," The Computer Journal, vol. 24, no. 2, pp. 167-172, 1981.

21. Yerry, M.A. and Shepard, M.S., "Automated Three-Dimensional Mesh Generation by the Modified Octree Technique," Int. J. Num. Method Eng., vol. 20, pp. 1965-1990, 1984.

22. Lohner, R. and Parikh, P., "Three-Dimensional Grid Generation by the Advancing Front Method," Int. J. Num. Meth. Fluids, vol. 8, pp. 1135-1149, 1988.

23. Hufford, G.S. and Mitchell, C.R., "The Generation of Hybrid and Unstructured Grids Using Curve and Area Sources," AIAA-95-015, Jan. 1995.

24. Coirier, W.J. and Powell, K.G., "A Cartesian Cell-Based Approach for Adaptively Refined Solutions of Euler and Navier-Stokes Equations," AIAA-95-0566, Jan. 1995.

25. Marcum, D., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA-95-0212, Jan. 1995.

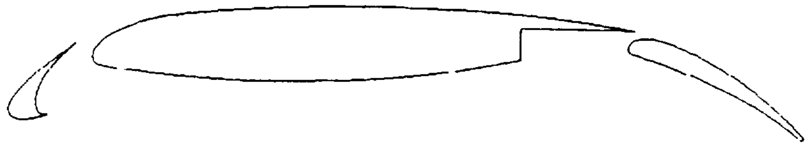

Figure 1. The base configuration of the 3-element airfoil case

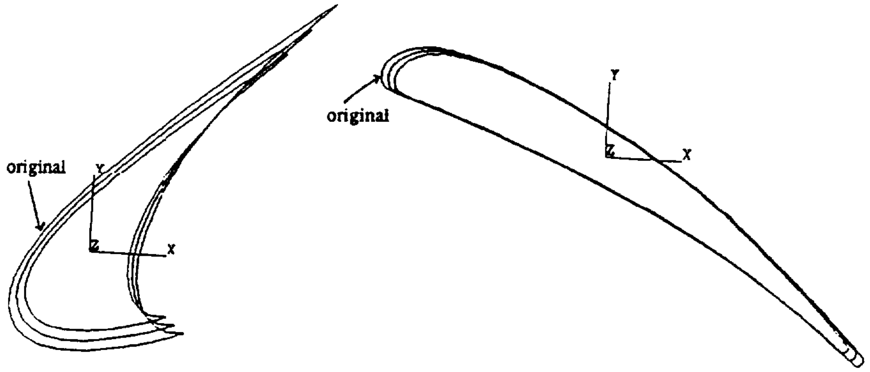

Figure 2. The scaled versions of the slat and flap parts compared to the original

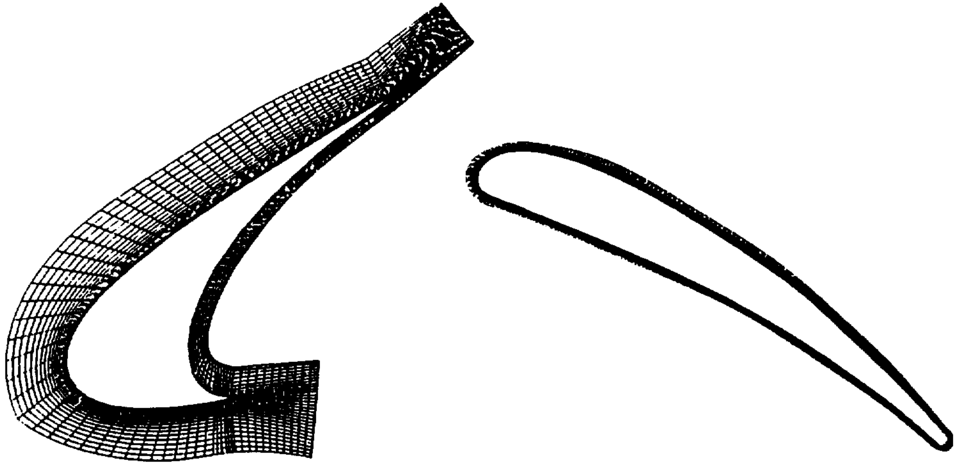

Figure 3. Structured grid surrounding the slat and flap parts. Both geometry and grid are saved in a library.

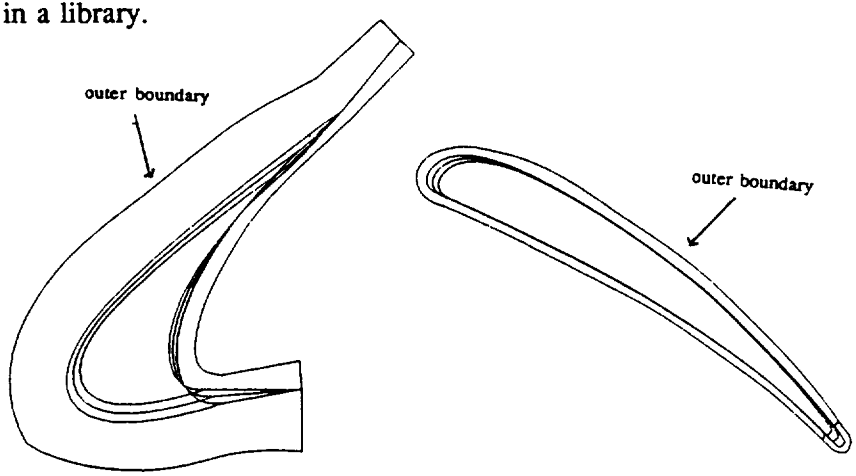

Figure 4. The common outer boundary for each series of slat/flap parts

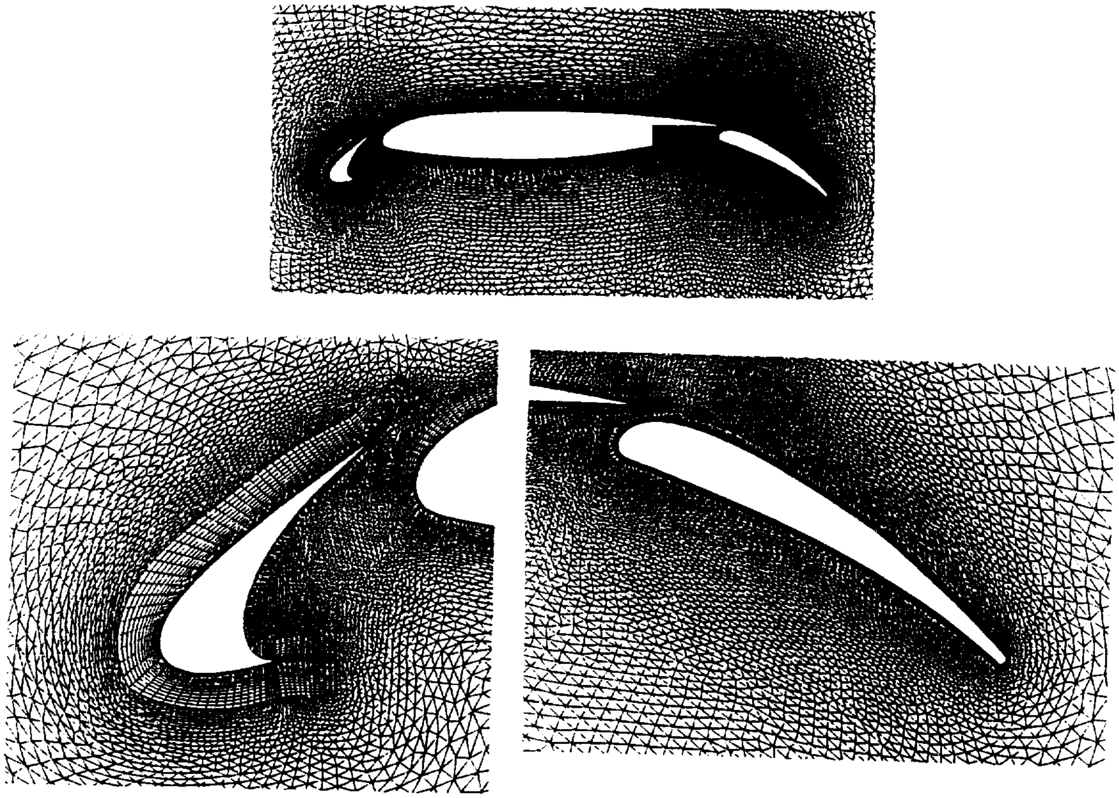

Figure 5. The unstructured grid that fills in the gaps between structured grid parts

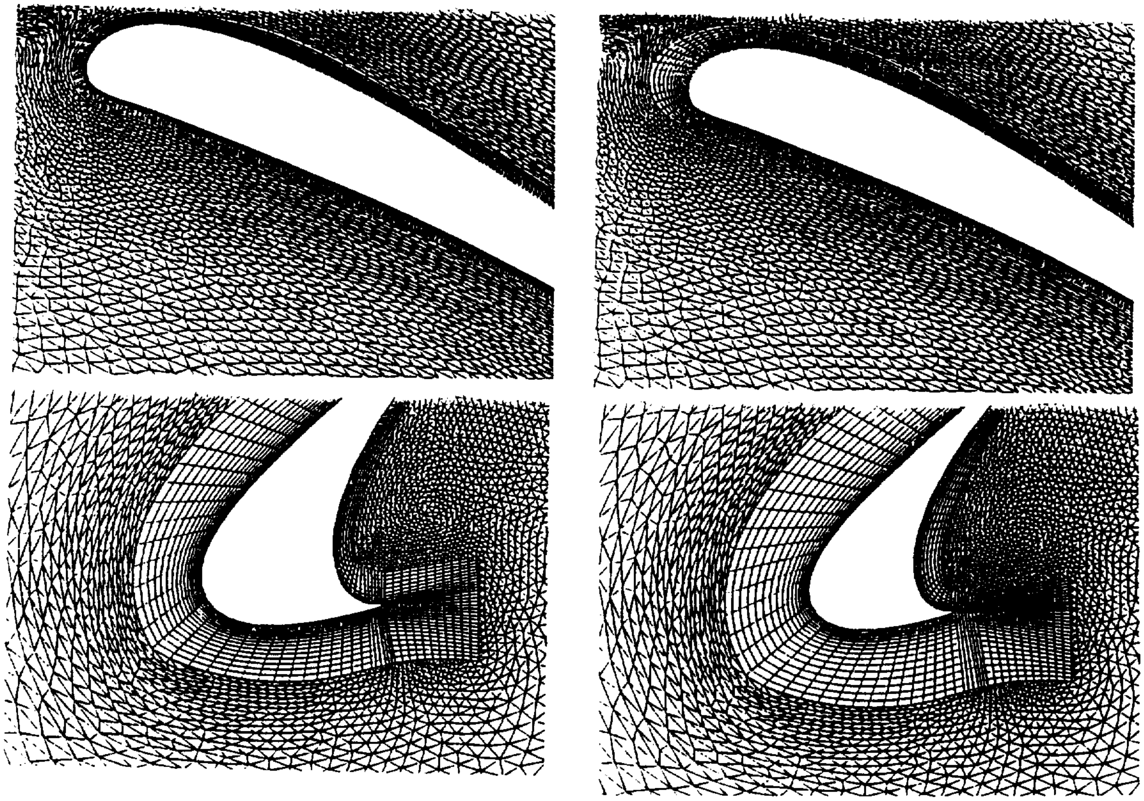

Figure 6. Interchanging of structured flap and slat grids without re-computation of unstructured grid

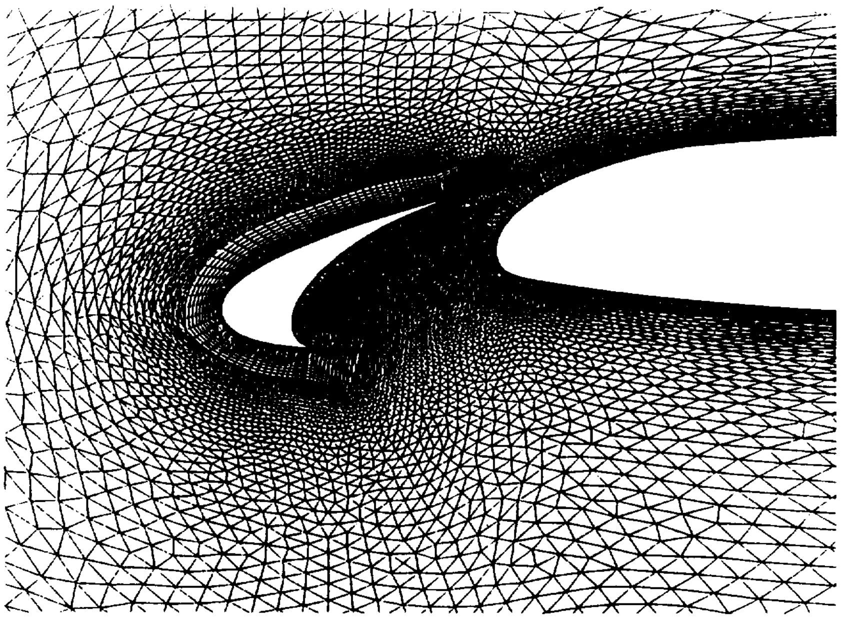

Figure 7. Example of slat rotation and unstructured grid re-computations with no additional user input

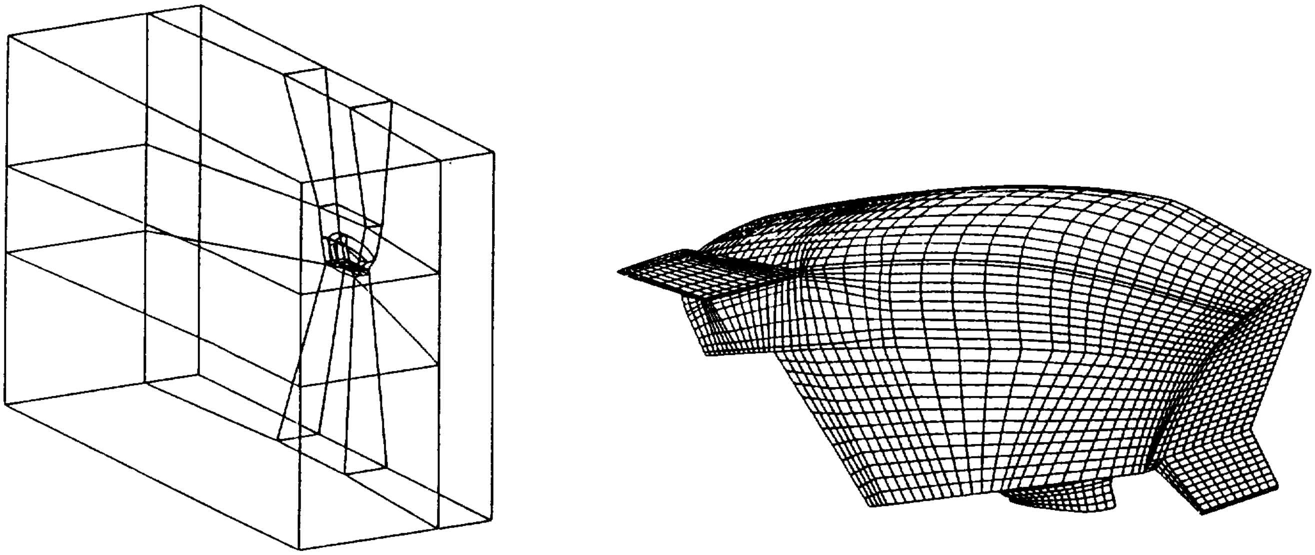

Figure 8. Multi-block topology of half of the B-1A escape capsule and the surface grid

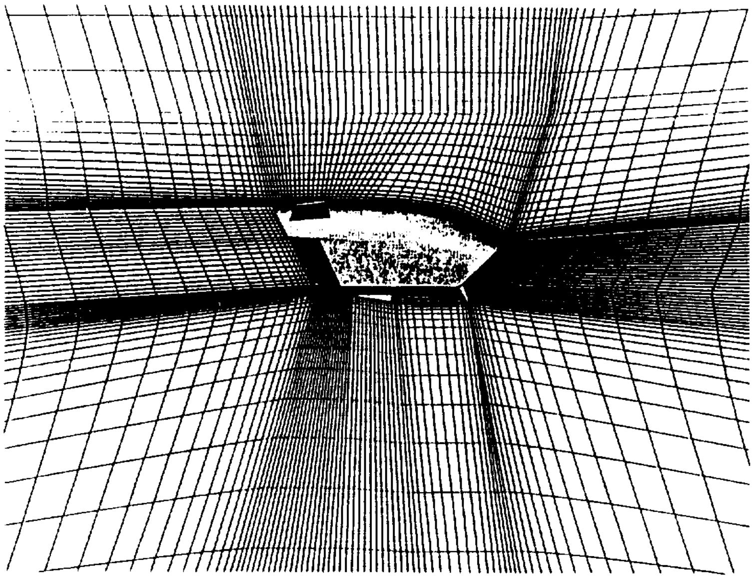

Figure 9. Algebraic grid of the B-1A capsule centerline

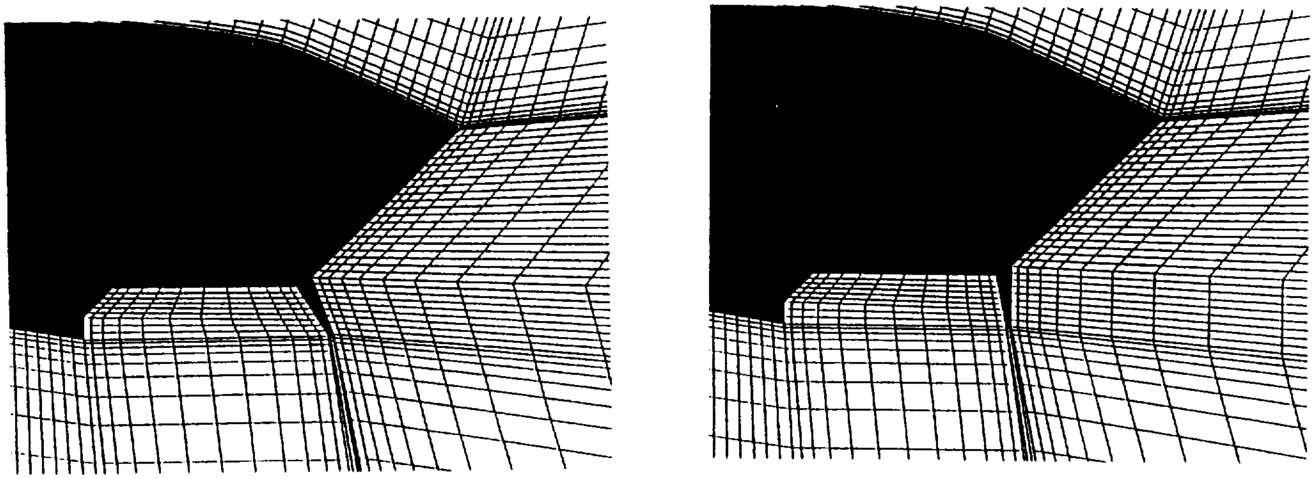

Figure 10. An example of a geometry modification (spoiler rotation) and immediate grid update using CFD-GEOM's structured gridding capabilities

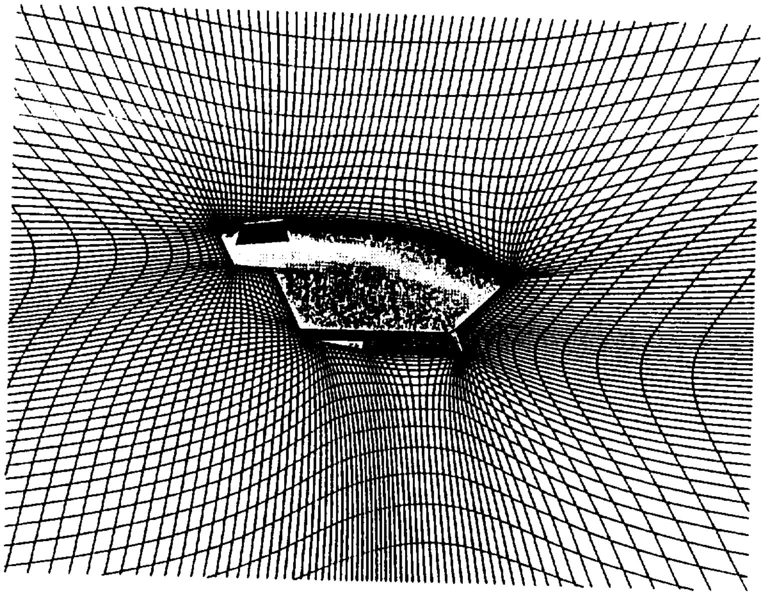

Figure 11. Elliptic grid at the B-1A capsule centerline

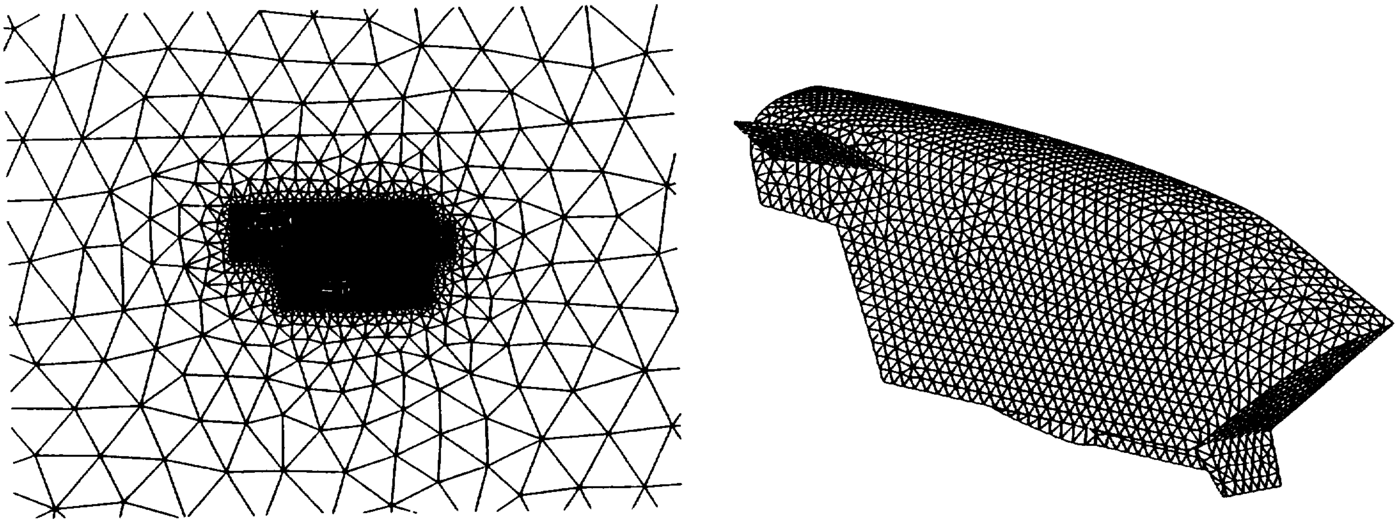

Figure 12. Unstructured grid of the B-1A escape capsule

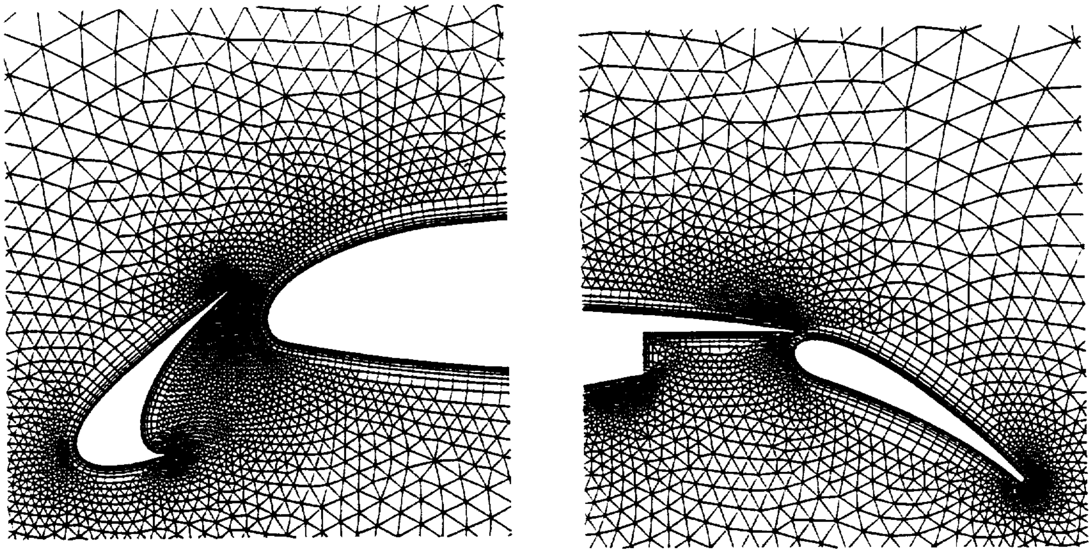

Figure 13. Type of grid produced from an advancing layer/normal type grid generator[25]

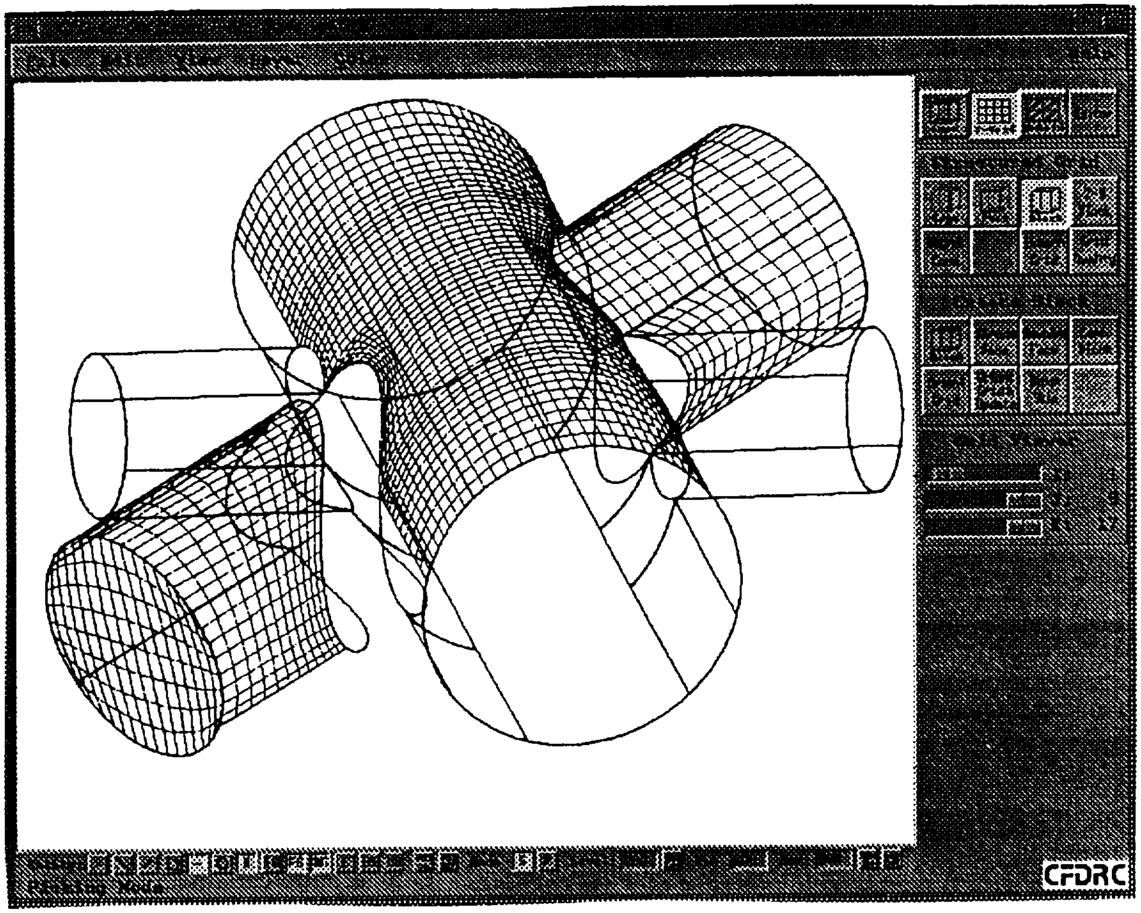

Figure 14. CFD-GEOM interface with a sample geometry and grid

# SOFTWARE SYSTEMS (1)