AN UNSTRUCTURED-GRID SOFTWARE SYSTEM FOR SOLVING COMPLEX AERODYNAMIC PROBLEMS

Neal T. Frink
NASA Langley Research Center, Hampton, Virginia 23681

Shahyar Pirzadeh and Paresh Parikh[1]
ViGYAN, Inc., Hampton, Virginia 23666

SUMMARY

A coordinated effort has been underway over the past four years to elevate unstructured-grid methodology to a mature level. The goal of this endeavor is to provide a validated capability to non-expert users for performing rapid aerodynamic analysis and design of complex configurations. The Euler component of the system is well developed, and is impacting a broad spectrum of engineering needs with capabilities such as rapid grid generation and inviscid flow analysis, inverse design, interactive boundary layers, and propulsion effects. Progress is also being made in the more tenuous Navier-Stokes component of the system. A robust grid generator is under development for constructing quality thin-layer tetrahedral grids, along with a companion Navier-Stokes flow solver. This paper presents an overview of this effort, along with a perspective on the present and future status of the methodology.

INTRODUCTION

The role of advanced Computational Fluid Dynamics (CFD) is becoming increasingly important in the aerodynamic design process. The transition of CFD technology from the research lab to the project environment has progressed rapidly in recent years with encouraging results. However, the intense competitive pressures on U.S. industries to build more efficient or specialized products, e.g. aircraft, automobiles, buildings, pumps, heat exchangers, etc., in less time and at lower cost and risk continue to challenge the most ardent CFD method developer. The challenge now is to complete the transition and realize the full potential of CFD technology in the engineering environment. The driving issues affecting CFD utilization in such an environment are time, confidence, cost, and risk [1]. Thus, codes to be applied in the project environment are those which are validated, robust, readily usable by non-CFD expert engineers, and provide reliable, timely information at a reasonable cost.

The unstructured-grid (USG) approach has moved us much closer toward meeting these challenges. This methodology now enables modestly trained engineers to generate tetrahedral grids on complex configurations in a matter of days [2]. Inviscid flow solutions are readily obtainable from a number of good unstructured flow solvers [3]. Significant progress is also being made toward the grid generation and flow solution for viscous-flow problems [3], but additional work is needed before a reliable viscous USG capability emerges. This paper will include the status of our progress in this area.

The status of USG Euler methodology is fairly mature and can play a significant role in the preliminary design phase of aerodynamic products. This inviscid technology has proven to be quite valuable as an engineering 'tool' for providing critical insights which can lead to the timely solution of complex aerodynamic problems. For example, Ref. [4] describes a successful effort to redesign the wing/pylon intersection junction of a transport aircraft with under-wing nacelles by reducing the inviscid peak suction pressures in the region prone to viscous flow separation. The resulting drag reduction was verified by flight test. Work is also underway to assess the utility of this technology for modeling the underhood flow of automobiles.

1. Consultant

The utility of USG Euler technology has been further enhanced by coupling it with other established engineering tools. As will be described in the text, an inexpensive 'viscous' capability is provided by coupling the flow solver with an interactive boundary-layer method. A design capability is available with the addition of a powerful inverse design algorithm. Propulsion effects can be included through special boundary conditions. And while not discussed in this paper, work is underway to link a USG flow solver and grid generator with an aeroelastic module. When complete, an engineering capability should be available for the rapid design of aerodynamic surfaces for applicable complex configurations, which include the effects of propulsion flow, aeroelastic deformation, and a boundary layer.

There has been an ongoing coordinated effort at NASA LaRC over the past four years to mature the USG methodology. The goal is to provide a validated capability to non-expert users for performing rapid aerodynamic analysis and design of complex configurations. This has been accomplished by advancing the methodology in a team environment using an application-oriented approach where the method development/calibration was guided by parallel focused application assessments from selected users within both NASA and industry. The remainder of this paper will present an overview of our system, and a perspective on the present and future status of the technology.

OVERVIEW OF CURRENT CAPABILITY

A diagram of the subject flow analysis system is presented in Figure 1. The primary components consist of a graphic user interface (GRIDTOOL), an advancing-front grid generator (VGRID), a cell-centered, upwind flow solver (USM3D), and a graphic post-processing code (VPLOT3D).

Grid Generation

GRIDTOOL.—GRIDTOOL [5] is a graphic user interface (GUI) utility developed to serve as a "window" into the unstructured-grid generator, VGRID. The user must provide a surface definition in one of several formats, i.e. IGES, GRIDGEN, PLOT3D, LAWGS, or simple networks, which are converted within GRIDTOOL into a NURBS surface for manipulation. This GUI provides easy access to many functions, to be described below, which are useful in setting-up an input file for VGRID.

VGRID.—VGRID [2,6] is an interactive program for generating unstructured, tetrahedral/triangular grids by the advancing-front method [7]. A grid is generated by forming cells starting from the domain boundaries marching towards the interior of the computational domain. The advancing-front process is illustrated in Figure 2 for a simple 2D domain. The configuration of interest is first defined in terms of a number of surface patches (line segments in 2D). A transparent 'background grid' is used to control the distribution of points. The boundary segments are then triangulated (divided) individually to form the surface mesh or initial front. Next, tetrahedral (triangular) cells are generated over faces on the front by introducing new points in the field. During this process, old faces are replaced by new ones (reshaping the front), and the front is advanced in the field until the whole region is filled with grid cells.

For a 3D configuration, the geometry is divided into subdomains, using GRIDTOOL, on which 3- or 4-sided parametric surface patches are defined. The process of surface triangulation within VGRID involves transforming the 3D patch into a 2D parameter plane while preserving the overall projected shape of the 3D patch. After surface triangulation in 2D, the newly generated points are then transformed back onto the 3D patch by interpolating from its boundary curves. Depending on the patch topology, it is possible for points not to return exactly to the original surface after the transformation. As noted in Figure 1, the surface grid can optionally be passed through GRIDTOOL to project the points onto the surface.

The distribution of grid points is controlled by a 'structured' background grid [8]. This transparent grid consists of Cartesian mesh overlaying the entire domain upon which the user prescribes 'point' and 'line' sources to impose the desired spacing distribution. Parameters are available to control cell size, and the direction and intensity of spatial variation. A smooth variation of spacing is achieved throughout the computational domain by solv-

ing an elliptic partial differential equation on the Cartesian mesh. The approach is analogous to modeling heat diffusion from discrete heat sources in a conducting medium. A smooth variation of cell spacing is critical to the robustness of the grid generator and the quality of the grid.

The volume grid can be generated on a workstation through a series of restarts with VGRID. The code exploits a recurrent local/global renumbering system [2] which drastically reduces the computer memory requirement. In each run, a prescribed number of grid points and cells are generated. With the renumbering system, the point and cell numbers, in each restart run, start from one rather than the accumulated global numbers. This allows specification of relatively small array dimensions regardless of the final grid size.

After VGRID has completed the volume grid, there are usually some distorted tetrahedra, or some voids in the domain where the grid generator could not close the front. These problems occur when fronts approach from different directions in such a way that an ideal next point cannot be identified. As noted in Figure 1, the generated volume grid is postprocessed by the code POSTGRID [2] which removes distorted cells and/or a layer of cells around the problem areas, creates new fronts, and resumes the advancing-front process to close the grid. Since the modified front is now smoother after removing the anomalies, the problems are usually corrected, and a better quality tetrahedral mesh is produced.

The new versions of VGRID and POSTGRID incorporate recent improvements for better grid quality, robustness, and easier grid generation. The enhancements include improved graphics for better visualization of interactive surface and volume grid generation, graphic version of POSTGRID, interactive check and improvement of the surface and volume grid quality, dynamic memory allocation, better CPU efficiency, and many other improvements. VGRID requires about 80 bytes of memory per tetrahedron, and generates about 220 tetrahedra (including I/O) per second using a Silicon Graphics Indigo^2 workstation with R4000 processor.

Flow Solver

USM3D.—USM3D [9-11] is a cell-centered, finite-volume upwind flow solver for solving the Euler equations on tetrahedral grids. Inviscid flux quantities are computed across each cell face using the Roe [12] flux-difference splitting approach (FDS), or the Van Leer [13] flux-vector splitting technique (FVS). Spatial discretization is accomplished by a novel cell reconstruction process, which is based on an analytical formulation for computing solution gradients within tetrahedral cells. Solutions are advanced in time by either a 3-stage Jameson-style Runge-Kutta explicit time-stepping scheme with convergence accelerated to steady state by local time stepping and implicit residual smoothing, or the linearized backward Euler implicit scheme of Ref. [14].

Cell reconstruction scheme: The higher-order reconstruction scheme, derived in Ref. [11,15] and illustrated in Figure 3, is based on a Taylor series expansion of the cell-averaged solution to the cell face. A key component of the scheme is the reconstruction of surrounding cell-averaged data to a common vertex or node by a weighted averaging procedure. Reference [9] proposed a scheme based on an inverse-distance weighted averaging of the primitive variables from the cell centroid to the cell vertices. While this approach has proven to be both accurate and robust through wide application, it is not fully second-order accurate in space. It has been shown in Ref. [16] to be approximately 1.85-order accurate.

As development efforts progressed toward solving the Navier-Stokes equations on highly stretched tetrahedral grids, it became evident that the accuracy of the inverse-distance averaging scheme was not adequate. Thus, a fully second-order accurate averaging procedure was implemented which is based on work by Holmes and Connell [17] and Rausch, et. al. [18]. The procedure is derived by solving a constrained minimization problem to determine weight factors which satisfy Laplacian relationships. The algorithm reconstructs to machine accuracy the exact values of a linear function at a node from surrounding cell-centered function values on an arbitrary tetrahedral grid. Furthermore, the simple universal formula shown in Figure 3 for expanding the cell-centered data to the cell faces also reconstructs the exact value of a linear function to the cell face. Thus, the entire spatial reconstruction scheme is termed second-order accurate, which has been verified by Mitchell [16].

Limiter device: Euler, finite-volume, upwind methods occasionally encounter circumstances where some local cell pressures, p_{cell}, become negative and preclude convergence of the global solution. This often occurs in regions where the inviscid assumptions are violated and viscosity would normally introduce enough diffusion to relieve the problem in real flow. Since the Euler equations are inviscid, the only source of diffusion in the solution comes from the discretization error inherent in the numerical approximation. A special empirical technique has been developed to resolve these local anomalies as they occur and permit the global solution to advance toward convergence. During each time iteration cycle, p_{cell} is not permitted to become less than a user prescribed minimum value, p_{tol}, i.e. $p_{cell}=max(p_{cell},p_{tol})$. Furthermore, for higher-order differencing, cells which satisfy $p_{cell}<p_{tol}$ are set to and maintained with a more diffusive 1st-order stencil by imposing $\Phi_{cell}=0$, as shown in Figure 3. In many cases, this technique introduces enough local numerical diffusion to stabilize the global solution and permit convergence. It has greatly increased the robustness of USM3D for complex geometries.

Data structure: An efficient face-based data structure [19] is exploited in the flow code, which does not require interface fluxes to be stored during the solution cycle. The flux algorithm cycles through the triangular faces and accumulates the residuals by scattering each flux to the two adjacent tetrahedral cells. For proper vectorization, face-based data structure requires that the faces be ordered into groups or "colors" in which no two faces share the same cell or opposing node. The explicit time integration scheme requires 45 words/cell of memory and 18 μs/cell/cycle (higher order) on a Cray Y-MP supercomputer. Similarly, the implicit scheme requires 180 words/cell and 64 μs/cell/cycle (higher order) on the Cray Y-MP. While the implicit scheme requires more CPU time per iteration than the explicit scheme, it is demonstrated in Ref. [11] that the implict algorithm converges in 6-times less computer time than the explict one since higher Courant-Friedrichs-Lewy (CFL) numbers can be utilized. The following table shows a machine-to-machine comparison of Central Processing Units (CPU) requirements for USM3D on a test grid for the ONERA M6 wing. The computations involved a combination of 1st- and higher-order differencing. No special effort was made to enhance vectorization or performance on the non-Cray machines. The larger relative CPU time for the Iris Indigo^2 between implicit (98.9) and explicit (27.6) time integration is related to excessive swapping of the buffer due to insufficient core memory. Implicit calculations were not made on the Sun 4 because of insufficient memory.

Computer	Explicit		Implicit	
	CPU time, μs/cell/cycle	CPU time relative to Cray Y-MP	CPU time, μs/cell/cycle	CPU time relative to Cray Y-MP
Cray C-90	7.0	0.44	29.0	0.45
Cray Y-MP	16.2	1.0	64.0	1.0
Convex	331.5	20.5	1234.0	19.3
Iris Indigo^2 (R4000)	448.0	27.6	6335.0	98.9
Sun 4	695.0	42.9	——	——

Boundary Conditions.—The USM3D code is designed for the easy addition/modification of boundary conditions (B.C.). It contains standard B.C.'s for flow tangency or no-slip on solid surfaces, characteristic inflow/outflow for subsonic outer boundaries, and freestream inflow and extrapolation outflow for supersonic flow. Some additional special boundary conditions, which have been added to improve the robustness of the code or to expand its capability, are described in the following.

Blunt base: With the application of USM3D to a wide range of complex 3D configurations, numerical difficulties routinely occur in blunt-based regions where a wake-type flow would develop in real flow, e.g. a blunt wing trailing edge or a blunt termination of a fuselage. For such cases, the inviscid flow assumption yields a singularity where the flow attempts to negotiate the corner.

A special boundary condition has been developed to mimic the relieving effect of a blunt-base wake on the surrounding inviscid flow field, which is to provide a smooth departure of the flow past the corner. As illustrated in Figure 4, the reconstructed primitive variables, (ρ, u, v, w, p), from the nodes of blunt-base boundary cells are averaged and assigned directly to the boundary face and ghost cell. The effect is to introduce a *solution defined* transpiration boundary condition to the subject faces which simulates the presence of a wake.

Engine model: USM3D contains a feature for modeling propulsion-induced effects of up to four jet-engine or jet-nozzle components. The model [20] computes inflow and/or outflow boundary conditions from user prescribed values for the total pressure and temperature change across or within the combustor, the exit pressure, the fuel fraction, and the direction cosines for the jet. When applicable, the inlet mass flow is automatically balanced with that of the nozzle.

Preprocessing.—A preprocessing step is involved before running USM3D on a given grid generated by VGRID. The VGRID code outputs a coordinate file of the vertex locations, a cell-to-node connectivity file, and a triangle-to-patch correlation file. The PREFLO code, depicted in Figure 1 generates an additional face-to-cell connectivity file which is required for the face-based data structure in USM3D. (There are plans to modify the VGRID output to generate the appropriate files and eliminate this preprocessing step.) Boundary conditions are applied at the time of USM3D execution through a B.C. map file output by GRIDTOOL.

Graphic Postprocessing, VPLOT3D

VPLOT3D is an interactive, menu-driven post processing program for extraction and display of data on unstructured (tetrahedral) grids. The philosophy behind the development of VPLOT3D has been to keep the program as simple and straight-forward as possible. There is only one 'text' window and only one main 'graphics display' window. Most of the interaction and steering takes place through pop-up menus.

VPLOT3D has the ability to display grids and flow quantities on either boundary surfaces or user defined arbitrary planes in the field. Flow quantities can be displayed using line or filled contours, velocity vectors and particle traces. Some unique features of the program include dynamic memory allocation, a 'probe' option that lets a user query the local value of a certain displayed fluid dynamic quantity, ability to plot two different data sets side-by-side and a journaling (log file) capability. Grid post-processing options allow a user to isolate and display 'bad' grid cells and to visually confirm boundary conditions before proceeding for flow solution. Several hardcopy output options are available including Black&White and Grayscale Postscript and RGB formats.

TECHNOLOGY ASSESSMENT

The present USG flow analysis system is being widely used throughout industry and government laboratories. Studies have been published related to subsonic transports [4,21], high-speed research [22,23], military aircraft technology [24], turbomachinery applications [25,26], and code calibration/assessments [27-29]. Results from a published code calibration study are included herein to illustrate the capability.

A study was performed to assess the accuracy and utility of VGRID/USM3D on a wing/pylon/finned store configuration [27] shown in Figure 5. Tetrahedral grids were generated for the configuration with baseline (or store-in-carriage) store position, two dropped positions for $M_\infty=0.95$, and two positions for $M_\infty=1.2$. The initial grid was generated prior to the development of GRIDTOOL and took 4 days to complete. (The same task would require 1 or 2 days with present codes.) The remaining four grids were generated by 'freezing' the baseline surface triangulation while translating/rotating the surface patches, background-grid sources, and surface grid for the store to prescribed locations, then generating a new volume grid. The latter four grids required approximately one-hour each to generate.

Solutions were computed with Runge-Kutta time stepping and convergence acceleration using a CFL number of 4 in about 2000 iterations, while the L2 norm (rms average of all residuals) decreased by about 4 orders

of magnitude. For the baseline position, the computation required about 5.7h of Cray Y-MP time, whereas for the other two positions, the required computational time was of the order of 4 CPU h.

Figure 5 compares computed and experimental C_p data [30] on the store along two streamwise rays (along top, $\phi=5^o$, and bottom, $\phi=185^o$) from the nose to the tail. For each case, the computed pressure coefficient compares very well with experimental data. In particular, note the comparison along the 5-deg ray for the baseline configuration that passes through the gap between the store and the pylon and where a large interference is expected. While not shown, very good agreement was also obtained on the fins of the store.

STATUS OF EMERGING CAPABILITY

Thin-Layer 'Viscous' Grid Generator

Advancing-Layers Method.—A new method for generation of thin-layer 'viscous' grids has been recently introduced and successfully applied to 2D [31] and 3D [32] problems. The method, referred to as the advancing-layers method (ALM), is based entirely on a modified advancing-front method (AFM) and benefits from the generality and flexibility of the conventional advancing-front-based Euler grid generators. Being based on an unstructured grid strategy, the method alleviates the structural limitations of the structured and semi-unstructured techniques.

The generation of a grid with the present approach is divided into three separate stages: 1) surface grid generation, 2) construction of high-aspect-ratio cells in the viscous region, and 3) generation of regular (isotropic) cells in the inviscid-flow region. Steps 1 and 3 utilize established methodology encompassed in the advancing-front inviscid grid generation code VGRID. The second step is performed independently by the ALM to reduce the overall complexity of the process. Although the generation of grid is divided into separate stages with the present approach, the entire process is performed in a single run with automatic transition from one stage to another.

The main features of the advancing-layers technique are similar to those of the conventional AFM. A volume grid is generated through a marching process in which tetrahedral cells originate from a triangular surface mesh and proceed into the computational domain. In contrast to the conventional method which adds cells in the field in no systematic sequence, the ALM advances one layer of cells at a time to reduce the complexity of producing high aspect ratio cells. Grid cells are formed by connecting new points, inserted along predetermined surface vectors using a prescribed stretching function, to the corresponding faces on the front. During the process, the integrity of the generated grid is monitored at each step. The layers continue to advance in the field, while growing in thickness, until either 1) opposite fronts approach to within a local cell size or 2) certain grid quality criteria, dictated by a global background grid, are locally satisfied. When the conditions are met on all faces on the front, the process automatically switches from the advancing-layers to the advancing-front mode to generate equilateral cells in the remaining inviscid-flow region.

To demonstrate the capability of the method, sample 2D and 3D grids around complex multi-component configurations are presented. The 2D geometry consists of three airfoil sections in a high-lift arrangement. The generated grid, shown in Figure 6, contains 34,987 cells and 17,705 points with a first layer spacing of about $7X10^{-6}$ main airfoil chord length. The flow solution, shown in Figure 6 was obtained on the grid at $M_\infty=0.2$, $a=16.2^o$, and $Re_{\bar{c}}= 9X10^6$ using an available node-based, upwind flow solver [33] with the Baldwin-Barth turbulence model.

A representative complex 3D configuration was constructed to show the ability of the method to negotiate difficult aspects of 3D viscous grid generation. Although the geometry resembles a quasi-two-dimensional shape, it contains 3D complexities such as multiple components, sharp convex and concave edges, small gaps, and thin surfaces at the wing tip. The generated grid (surface and volume) is shown in Figure 7. The 'viscous' portion of the

grid contains 490,266 tetrahedral cells and has been generated using an SGI Indigo^2 workstation (R4000 processor) in only 4.3 minutes. The final grid contains a total of 168,734 points and 958,716 cells.

Anisotropic stretching:—In order to generate viscous grids of reasonable size on complex configurations, it will be necessary to have anisotropic (stretched) surface grids so that fewer points can be distributed in directions of reduced flow field gradients. The development of such a capability is currently under way which has produced satisfactory preliminary results. Figure 8 compares two surface grids for a wing configuration: one isotropic and another stretched along the leading and trailing edges of the wing. Also the field grid distributions are represented by triangulating a plane passing through the middle of the wing. The isotropic surface mesh (wing only) contains 40,778 equilateral triangles and is shown in Figure 8(a). A corresponding stretched grid with a similar chordwise grid distribution is shown in Figure 8(b). As indicated, the grid is stretched along the edges of the wing in different directions, with the stretching propagated in the field close to the body and smoothly dissipated further away from the geometry. The maximum cell aspect ratio in this grid is about 20:1 at the leading edge of the wing. Although the stretched grid has a similar distribution of points in the chordwise direction, it contains only 6,841 triangles on the wing.

Navier-Stokes Flow Solver

Considerable progress has been made toward solving the Navier-Stokes equations on thin-layer tetrahedral grids with USM3D. Ref. [11] provided an initial validation of the solution algorithm on a tetrehedral grid constructed by subdividing the hexahedral cells of a structured grid. The following presents a full demonstration of computing a viscous-flow solution on a fully unstructured thin-layer tetrahedral grid produced from VGRID.

Hummel delta wing.—An unstructured viscous grid of 730,454 tetrahedral cells was generated for a delta wing of aspect-ratio 1 by an ALM version of the VGRID code. Normal spacing was distributed with geometric stretching near the surface to yield approximately 10 nodes (30 tetrahedra) in the mid-chord boundary layer for laminar flow. Anisotropic stretching, evident in the surface grid shown in the Figure 9, was utilized to reduce the chordwise density of cells in the grid—this feature resulted in a 7-fold reduction in grid size compared to standard isotropic triangles.

A viscous flow solution was computed with USM3D. A companion structured-grid computation was also performed with the CFL3D code [34] on a 65X65X33 (span-radial-chordwise) H-O grid for comparison. Both the unstructured and structured computations were performed on the delta wing at $M_\infty=0.3$, $a=20.5^\circ$, and $\mathrm{Re}_L=0.9X10^6$ with laminar flow.

The figure portrays the surface "oil-flow" pattern and a comparison of the spanwise distribution of pressure coefficient, C_p, at four chord stations. The coalescing and diverging streamlines in the computed flow patterns show evidence of the primary, secondary, and tertiary vortices. Good agreement is noted between the unstructured and structured solutions, and with the experimental data of Hummel [35].

Turbulence modeling.—The one equation turbulence model of Spalart-Allmaras [36] has been installed in USM3D and is presently undergoing testing. The derivation of this model relies heavily on empirical reasoning and dimensional analysis, and is receiving a growing acceptance within the CFD community. The turbulence equation solves for turbulent viscosity separately from the flow equations during each time step, resulting in a loosely coupled solution process. The governing equation is solved using the same backward-Euler time-stepping scheme [14] as used for the flow equations.

A preliminary assessment of the turbulence model has been made for flat-plate boundary layer flow. A two-dimensional flow was simulated by computing on a three-dimensional 'channel' grid which was constructed by subdividing the hexahedral cells of a structured grid into tetrahedra. The grid was sized for a $y^+\approx1$ and 20-25 nodes across the boundary layer with $M_\infty=0.5$ and $\mathrm{Re}_L=2X10^6$. Figure 10 depicts a very good agreement between the computed behavior of the law-of-the-wall and the model of Spalding [37]. Additional assessments are under-

way for a NACA 0012 airfoil and the ONERA M6 wing. The use of wall functions to eliminate the need to resolve the laminar sublayer is also under investigation.

Interactive Boundary Layer

An interactive boundary layer (IBL) capability has been installed into USM3D [38] to provide for an inexpensive accounting of viscous effects in an unstructured-grid flow solver for predominantly attached flow cases, such as the transonic cruise condition of a subsonic transport. The two-dimensional strip boundary-layer method of Cebeci et al. [39] for turbulent, compressible flow was integrated with USM3D in a coupled fashion where the displacement thickness was simulated by transpiration velocities. The automated process begins with a partially converged inviscid solution and continues with the boundary layer method being invoked at specified intervals throughout the remainder of the solution iterations. The inviscid pressures along chordwise strips on the wing surface are passed to the boundary layer routine which calculates the estimated boundary layer displacement thickness. The transpiration velocities are computed along each strip from the boundary layer displacement thickness using Lighthill's equation [40]. A linear interpolation is performed between the strips to establish transpiration velocities at the remaining surface nodes. The face-based displacement boundary conditions are determined by averaging the node-based transpiration velocities to the centroid of the boundary triangle, then modifying the original tangent-flow condition to include the non-zero normal velocity. Face-center density and energy are set equal to their corresponding cell-center value. A similar capability for a node-centered unstructured flow solver is described in Ref. [41].

An application of the unstructured IBL capability, taken from Ref. [38], is shown in Figure 11 for a low-wing transport configuration with a supercritical airfoil at $M_\infty=0.77$, $a=1.12^o$, and $Re_{\bar{c}}=2.6X10^6$. The computational grid contains 286,000 tetrahedral cells and has ten chordwise strips along the wing span for applying the boundary layer method. A comparison of pressure coefficient at the 34-percent semispan station is shown in Figure 11. The inclusion of the boundary layer model results in substantial improvement in predicting both shock location and strength, as well as in the suction plateau. The poor agreement between the Euler result and the experimental data is typical of inviscid solutions on supercritical airfoils, where the shock location can be overpredicted by as much as 10- to 20-percent aft-chord.

Inverse Iterative Design

The Constrained Direct Iterative Surface Curvature (CDISC) design method [42] has been coupled with USM3D to provide a capability for aerodynamic design of complex configurations. The CDISC method automatically generates and modifies the target pressures that are used by the basic DISC design module so that flow and geometry constraints are met. The CDISC module is coupled with the flow solver in an iterative manner and is called after a user-specified number of flow iterations (typically 25). After the target pressures are altered to meet the design constraints, the basic DISC module is used to modify the affected surface points. Geometry constraints are directly enforced on this new surface, then the volume grid is perturbed, using the spring analogy of Ref. [43], to reflect the geometry changes. Since the target pressure modification, surface design, and flow solution are converged in parallel, and the CDISC design system components themselves are very efficient, a converged design is usually obtained in about 25-50% more time beyond the initial baseline analysis.

To illustrate the USM3D/CDISC design capability, the wing of an executive transport configuration (Figure 12) has been redesigned for reduced drag at a primary design point without increasing the drag at a secondary design point. The baseline configuration was first analyzed using the USM3D code to establish reference values for performance comparisons at the primary (Mach = 0.8) and secondary (Mach = 0.9) design points. These conditions correspond to the long-range and high-speed cruise points, respectively. Two approaches are available in CDISC for reducing the wave drag of a configuration at transonic speeds: 1) a simple incremental method that limits the Mach number ahead of a shock to a specified value while matching lift and pitching-moment constraints; and 2) the Modified Uniform Distribution (MUD) method, which redistributes the chordwise loading over an airfoil section. The MUD option was selected for this case and was used to redesign the wing to reduce the drag at the pri-

mary design point. The wing planform, nacelle location, root airfoil section, airfoil maximum thickness, and spanwise load distribution were all held fixed. Initial results showed a 15 count drag improvement at Mach = 0.8, with only a 3 count penalty at Mach = 0.9. This design however had an unacceptable twist distribution just outboard of the nacelle. An input parameter to the MUD method was adjusted to shift the chordwise loading forward at several sections, which brought the twist distribution much closer to that of the original wing. The baseline and modified pressure distributions at Mach = 0.8, along with the corresponding airfoil sections, are shown in Figure 12 for two wing stations. At the inboard station, the strong shock near the quarter-chord of the baseline wing has been significantly weakened, resulting in an overall drag decrease of 6 counts. Analysis at the high-speed cruise point indicated that the new design had a 1 count drag improvement over the baseline at that condition. The design took an additional 4 hours of Cray C-90 time beyond the 16 hours required to obtain the initial baseline analysis. The grid had 700,000 tetrahedral cells, producing a central memory requirement of 148 megawords.

Solution Adaptive Gridding

One of the strong advantages of unstructured-grid technology is the potential to exploit its flexibility in adapting or clustering the grid in the vicinity of dominant flow features, such as shocks and vortices, to enhance solution accuracy with fewer cells. This capability should facilitate a more rapid transfer of the technology to the workstation environment. Attention has been focused on providing an unstructured "remeshing" capability which distributes grid points efficiently throughout the domain and adequately resolves important flow features.

Remeshing procedures have been developed and applied for shock [44] and vortex [45] dominated flows. The procedure integrates the grid generation and flow solver functions in such a way that the information produced by one is successively used by the other until a desired accuracy is obtained. Accordingly, the grid generator VGRID, is combined with the flow solver USM3D, to produce a unified procedure that produces input parameters for the mesh generator based on the solution on a previous grid. The concept of a background grid, which is used in the advancing-front technique to specify and control grid size distribution, is exploited in the remeshing procedure.

Shocked flows:—The procedure of Ref. [44], which addressed shock dominated flows, was based on the "tetrahedral" background grid[1] concept [7] where the computational grid serves as the background grid for generating a subsequent grid. Grid spacing parameters were calculated at the nodes of this mesh based on the previous flow solution, and a new mesh was generated. Figure 13 shows the result of applying this remeshing procedure on the ONERA M6 wing. The upper portion of the figure compares the upper surface triangulation between the adapted grid (at the end of three adaptation cycles) and an unadapted grid. On the lower portion, C_p distribution at two spanwise stations are compared (for $M_\infty=0.84$ and $a=3.06^\circ$) between the adapted and unadapted grids, and the experimental data. While both sets of computed data compare well with the experimental values, the efficiency of the adaptive procedure is established by the fact that the adapted grid had about 2.5 times fewer cells than the unadapted case.

Vortex flows:—A solution adaptive remeshing procedure, based on the "structured" Cartesian background grid concept, has been developed for low-speed vortex dominated flows [45], such as those produced by sharp swept-back leading edges or shed from the free ends of lifting surfaces. As discussed earlier with the structured background grid, grid spacing is controlled by point and line 'sources' strategically placed throughout the domain by the user. The present approach provides a capability for automatically prescribing line sources for refining the grid in regions of vortex flow.

In the current procedure, the vortices are located by an entropy parameter. In addition, a vortex in close proximity to a surface induces a static pressure peak on the surface. To accurately resolve both the vortex core and the accompanying static pressure peak, line sources are placed in the field and on the surface of the configuration. The procedure is automated to the point that the user need only specify the approximate starting locations for the

1. The "tetrahedral" background grid is no longer used in VGRID, but has been replaced by the "structured" background grid approach.

vortices and threshold cut-off values for both the entropy and the static pressure. All other functions, such as identifying the location and strength of the background grid 'sources' and creation of a new input for the grid generator, are automated.

The adaptation procedure has been applied to the Modular Transonic Vortex Interaction (MTVI) model, tested at NASA Langley, which is characterized by the presence of multiple vortices (see Figure 14). Computations were initiated on a relatively coarse grid with 35,388 points and 188,304 tetrahedral cells (GRID 1). Two successively finer adapted grids were generated with the adaptive procedure. The final adapted grid (GRID 3) had 68,158 points and 371,360 cells.

The left side of Figure 14 shows a comparison of surface triangulation for the unadapted (GRID 1) and the adapted (GRID 3) cases. For the adapted case, surface grid clustering near leading edges as well as near the wing tip is evident representing the static pressure peak due to proximity of vortices to the configuration surface. The bottom right portion of the figure shows a comparison of the field grid projected on to a cross sectional plane at 93-percent of root chord. Grid clustering around the vortex cores between the fuselage and the vertical tail and near the wing tip can be seen clearly. On the top right of the figure, flow field results are compared between the unadapted (GRID 1), the adapted (GRID 3) and experimental data. The C_p comparison is shown at a streamwise station located at 93-percent of the root chord for $M_\infty=0.4$ and $a=10.54^o$. Results are also compared from a 'fine' grid with 148,285 points and 825,469 tetrahedra . During generation of the 'fine' grid, no special effort was made to cluster grid points around vortex core locations. An expected higher suction peak resulting from grid adaptation is evident. The fine grid did not capture the suction peak at the station shown due to lack of grid resolution there, while the adapted grid automatically provided the grid resolution needed. This case clearly establishes the ability of the adaptation procedure to automatically provide adequate grid resolution where needed. The whole adaptation process required about 2.5 hours of CPU time on a Cray-YMP computer, compared to about 4.0 hours for the 'fine' grid. The adapted results show a better accuracy with about 2.2 times less number of cells and about 38-percent CPU saving compared to the 'fine' grid case.

CONCLUDING REMARKS

An overview has been presented of a maturing capability for performing rapid aerodynamic analysis and design of complex configurations using unstructured-grid methodology. Development priorities have been dictated by program needs from within NASA as well as those of industry partners. This technology also offers strong potential for impacting other non-aerospace disciplines, such as automobiles, boat sail and hull design, wind loads on buildings, design of ventilation systems, analysis of pumps and pipe systems, design of heat exchangers, and electromagnetics. The present unstructured Euler technology can be readily applied by modestly trained engineers to solve complex aerodynamic problems. Significant progress has also been made towards developing an unstructured Navier-Stokes capability for complex geometries.

Considerable utility can be derived from the base Euler technology. Often, the inviscid flow equations can provide the critical aerodynamic insights needed to resolve more complex problems. The Euler methodology has also been enhanced with additional engineering features such as interacting boundary layers, constrained inverse design, propulsion flow boundary conditions, and solution-adaptive gridding, thereby providing a powerful capability for aerodynamic design and analysis to the project-level engineer.

The capability for solving the Navier-Stokes equations on complex configurations using unstructured-grid methodology is still evolving. Such a flow solver has been developed and is presently being tested. The primary obstacle to a practical unstructured Navier-Stokes capability has been the lack of a robust thin-layer 'viscous' tetrahedral grid generator. As demonstrated in this paper, the Advancing-Layers method shows great promise toward providing that capability.

ACKNOWLEDGMENTS

Many of the new capabilities presented herein reflect the contributions of several talented people to whom the authors are profoundly grateful. **Dr. W. Kyle Anderson** of the Aerodynamic and Acoustics Branch at Langley provided generous assistance with installing his backward Euler time-implicit subroutines into USM3D, as well as providing technical advise on installing the 1-equation turbulence model in the code. **Dr. Peter M. Hartwich**, now with McDonnell Douglas Aerospace - West, integrated the engine model capability into the flow solver. **Ms. Leigh A. Smith** and **Mr. Richard L. Campbell** of the Transonic/Supersonic Aerodynamics Branch (T/SAB) at Langley contributed the work and text on the CDISC inverse-design method. **Mr. William D. Smith** and **Mr. Jassim A. Al-Saadi,** also of T/SAB contributed the IBL material.

REFERENCES

1. Cosner, R. R. : "Issues in Aerospace Application of CFD Analysis", AIAA Paper 94-0464, January 1994.
2. Pirzadeh, S. : "Recent Progress in Unstructured Grid Generation", AIAA Paper 92-0445, January 1992.
3. Venkatakrishnan, V. : "A Perspective on Unstructured Grid Flow Solvers", AIAA Paper 95-0667, January 1995.
4. Potsdam, M. A., Intemann, G. A., Frink, N. T., Campbell, R. L., Smith, L. A., and Pirzadeh, S.: "Wing/Pylon Fillet Design Using Unstructured Mesh Euler Solvers", AIAA 93-3500, August, 1993.
5. Samareh-Abolhassani, Jamshid, "GridTool: A Tool for Surface Modeling and Grid Generation", *Proceedings of Workshop on Surface Modeling, Grid Generation, and Related Issues in CFD Solutions,* NASA CP-3291, May 9-11, 1995.
6. Parikh, P., Pirzadeh, S., and Löhner, R. : "A Package for 3-D Unstructured Grid Generation, Finite-Element Flow Solutions, and Flow-Field Visualization", NASA CR-182090, September 1990.
7. Löhner, R., and Parikh, P. : "Three-Dimensional Grid Generation by the Advancing Front Method", *Int. J. Num. Meth. Fluids*, 8, 1988, pp. 1135-1149.
8. Pirzadeh, S. : "Structured Background Grids for Generation of Unstructured Grids by Advancing-Front Method", *AIAA Journal*, Vol. 31, No. 2, pp. 257-265, February 1993.
9. Frink, N. T. : "Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes", AIAA Journal, Vol. 30, No. 1, January 1992, pp. 70-77.
10. Frink, N. T., Parikh, P., and Pirzadeh, S. : "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes", AIAA 91-0102, January 1991.
11. Frink, N. T. : "Recent Progress Toward a Three-Dimensional Unstructured Navier-Stokes Solver", AIAA Paper 94-0061, January 1994.
12. Roe, P. L. : "Characteristic Based Schemes for the Euler Equations", *Annual Review of Fluid Mechanics*, Vol. 18, 1986, pp. 337-365.
13. Van Leer, B. : "Flux-Vector Splitting for the Euler Equations", *Eighth International Conference on Numerical Methods in Fluid Dynamics*, E. Krause, ed., *Volume 170 of lecture Notes in Physics*, Springer-Verlag, 1982, pp. 507-512.
14. Anderson, W. K. : "Grid Generation and Flow Solution Method for Euler Equations on Unstructured Grids", NASA TM-4295, April 1992.
15. Frink, N. T. : "Three-Dimensional Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Grids", Ph. D. Dissertation, Virginia Polytechnic Institute and State University September 1991.

16. Mitchell, C. R. : "Improved Reconstruction Schemes for the Navier-Stokes Equations on Unstructured Meshes", AIAA 94-0642, January 1994.

17. Holmes, D. G. and Connell, S. D. : "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids", presented at the *AIAA 9th Computational Fluid Dynamics Conference,* June 1989.

18. Rausch, R. D., Batina, J. T., and Yang, H. T. Y. : "Spatial Adaption Procedures on Unstructured Meshes For Accurate Unsteady Aerodynamic Flow Computation", AIAA Paper 91-1106, April 1991.

19. Jameson, A. and Baker, T. J. : "Improvements to the Aircraft Euler Method", AIAA Paper 87-0452, January 1987.

20. Hartwich, P. M. and Frink, N. T. : "Estimation of Propulsion-Induced Effects on Transonic Flows Over a Hypersonic Configuration", AIAA Paper 92-0523, January 1992.

21. Dodbele, S. S.: "Three Dimensional Aerodynamic Analysis of a High-Lift Transport Configuration", AIAA 93-3536, August, 1993.

22. Fouladi, K.: "Unstructured Grids for Sonic-Boom Analysis", AIAA 93-2929, July, 1993.

23. Carlsen, W. D.: "Development of Transonic Area-Rule Methodology", AIAA 94-1841, June 1994.

24. Wai, J., Herling, W. W., and Muilenburg, D. A., "Analysis of a Joined-Wing Configuration", AIAA 94-0657, January 1994.

25. Kwon, O. J., and Hah, C.: "Three-Dimensional Unstructured Grid Euler Method Applied to Turbine Blades", AIAA 93-0196, January, 1993.

26.Kwon, O. J. : "Solution of the 3-D Navier-Stokes Equations with a Two-Equation Turbulence Model on Unstructured Meshes Applied to Turbomachinery", AIAA Paper 94-1833, June 1994.

27. Parikh, P., Pirzadeh, S., and Frink, N. T. : "Unstructured Grid Solutions to a Wing/Pylon/Store Configuration", *Journal of Aircraft*, Vol. 31, No. 6, pp. 1291-1296, November-December 1994.

28. Ghaffari, F.: "On The Vortical-Flow Prediction Capability Of An Unstructured Grid Euler Solver", AIAA 94-0163, January, 1994.

29. Kinard, T. A., and Harris, B. W.: "Evaluation of Two Unstructured CFD Methods", AIAA 94-1877, June 1994.

30. Heim, E. R. : "CFD Wing/Pylon/Finned Store Mutual Interference Wind Tunnel Experiment", Arnold Engineering Development Center, AEDC-TSR-91-P4, January 1991

31. Pirzadeh, S. : "Unstructured Viscous Grid Generation by Advancing-Layers Method", AIAA Paper 93-3453, August, 1993.

32. Pirzadeh, S. : "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method", AIAA Paper No. 94-0417, January 1994.

33. Anderson, W. K. and Bonhaus, D. L. : "An Implicit Upwind Algorithm For Computing Turbulent Flows On Unstructured Grids", *Computers Fluids*, Vol. 23, No. 1, pp. 1-21, 1994.

34. Thomas, J. L., Taylor, S. L., and Anderson, W. K. : "Navier-Stokes Computations of Vortical Flows Over Low Aspect Ratio Wings", AIAA Paper 87-0207, January 1987.

35. Hummel, D. : "On the Vortex Formation Over a Slender Wing at Large Angles of Incidence", AGARD CP-247, Paper No. 15, February 1983.

36. Spalart, P. R. and Allmaras, S. R. : "A One-Equation Turbulence Model for Aerodynamic Flows", AIAA 92-0439, January 1992.

37. Spalding, D. B. : *Journal of Applied Mechanics*, Vol. 28, pp. 455-457.

38. Smith, William D. : "Improved Pressure and Lift Predictions in Transonic Flow Using in Transonic Flow Using an Unstructured Mesh euler Method with an Interacting Boundary Layer", Masters of Science Thesis, The George Washington University, July 1994.

39. Cebeci, T., Clark, R. W., Chang, K. C., Halsey, N. D., and Lee, K. : "Airfoils with Separation and the Resulting Wakes", *Journal of Fluid Mechanics*, Vol. 163, pp. 323-347, 1986.

40. Lighthill, M. J. : "On Displacement Thickness", *Journal of Fluid Mechanics*, Vol. 4, Part 4, pp. 383-392, 1958.

41. Potsdam, M. A. : "An Unstructured Mesh Euler and Interactive Boundary Layer Method for Complex Configurations", AIAA Paper 94-1844, June 1994.

42. Campbell, Richard L : "An Approach to Constrained Aerodynamic Design With Application to Airfoils", NASA TP 3260, November, 1992.

43. Batina, J. T. : "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex Aircraft Aeroelastic Analysis", AIAA 89-1189, 1989.

44. Parikh, P. and Frink, N. T. : "An Adaptive Remeshing Procedure for three Dimensional Unstructured Grids", *4th International Symposium on Computational Fluid Dynamics*, Davis, CA, September 1991.

45. Parikh, P. : "An Adaptive Remeshing Scheme For Vortex Dominated Flows Using Three Dimensional Unstructured Grids", ICASE/LaRC *Workshop on Adaptive Grid Methods*, Hampton, VA, November 7-9, 1994. (To be published in NASA Conference Proceedings.)

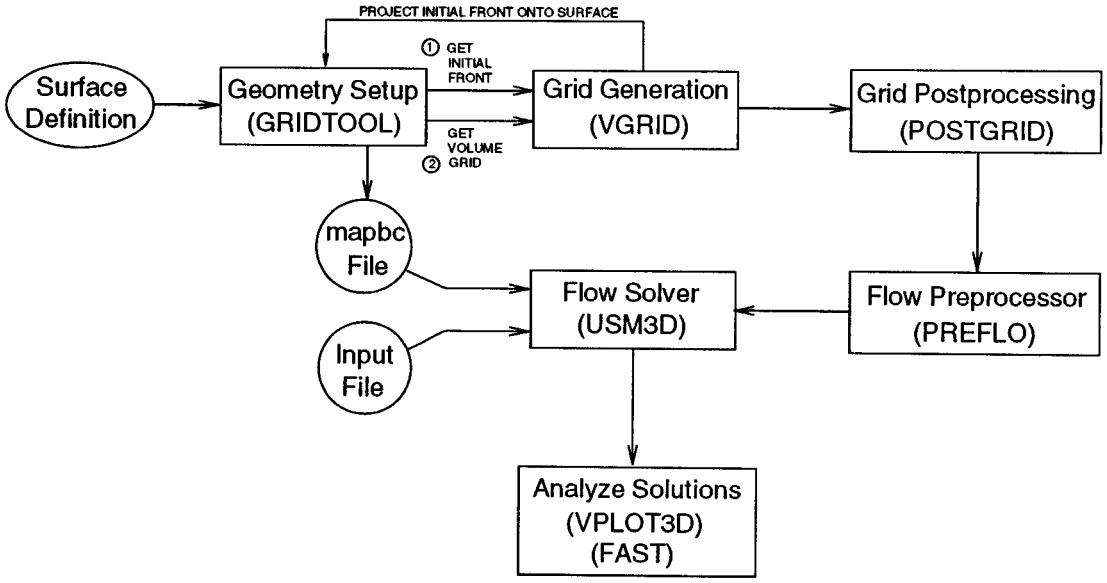


Figure 1. - Flowchart of unstructured-grid flow analysis system.

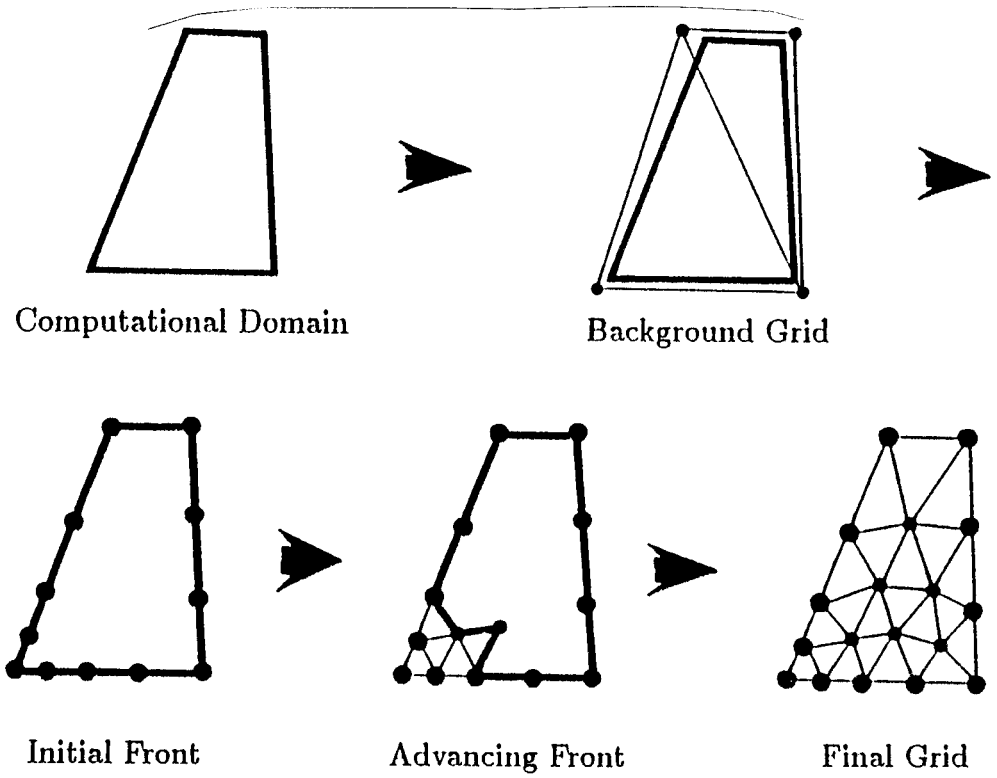


Figure 2. - Advancing-front process for a simple 2D domain

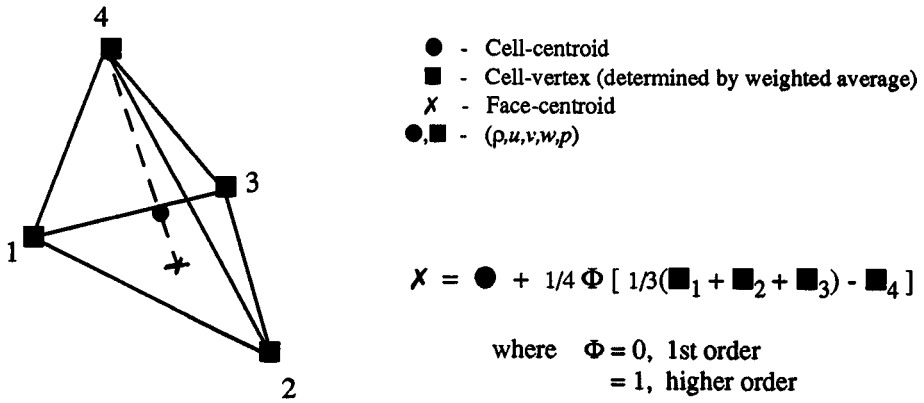


Figure 3. - Higher-order stencil for tetrahedral cell-centered scheme.

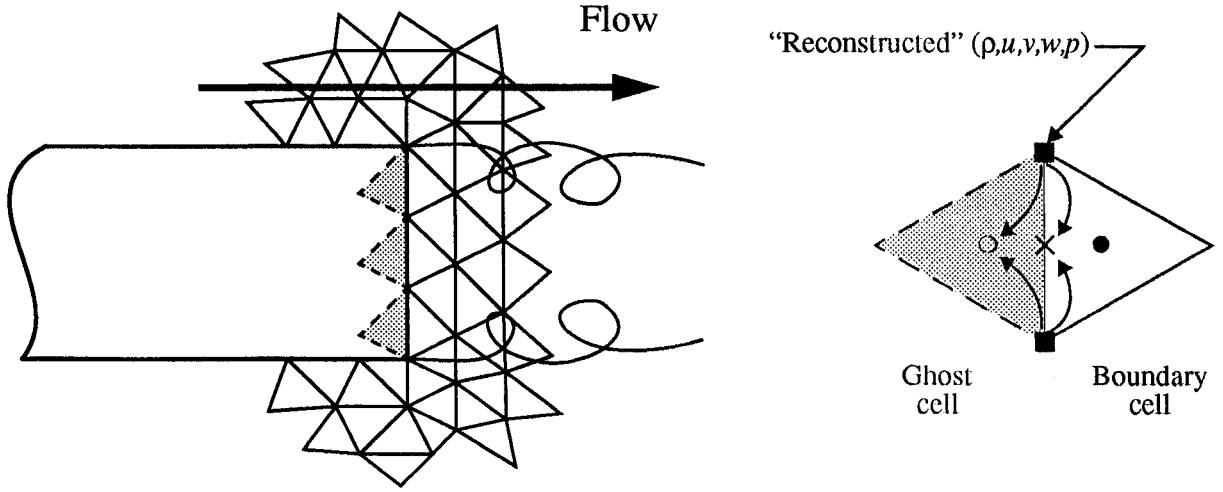


Figure 4. - Schematic of blunt-base boundary condition

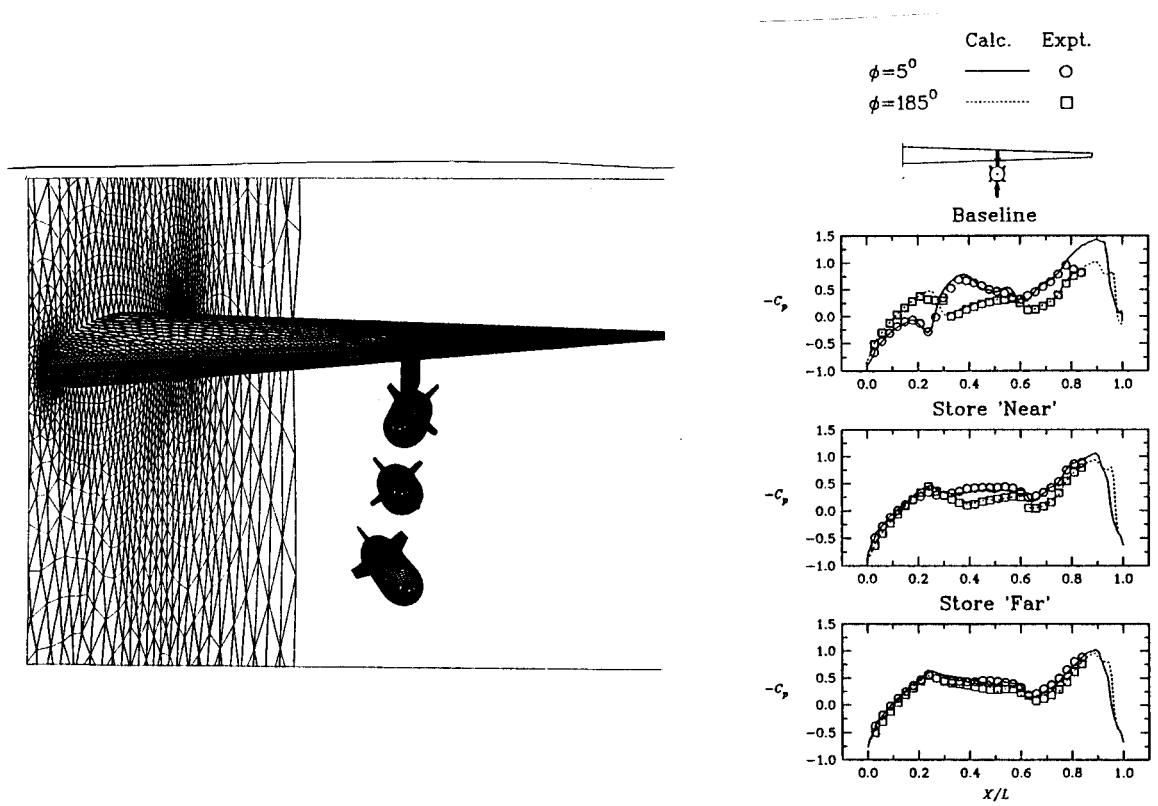


Figure 5. - - Surface triangulation on a wing/pylon/store configuration, and surface pressure comparisons on store along streamwise rays, M_∞=0.95.

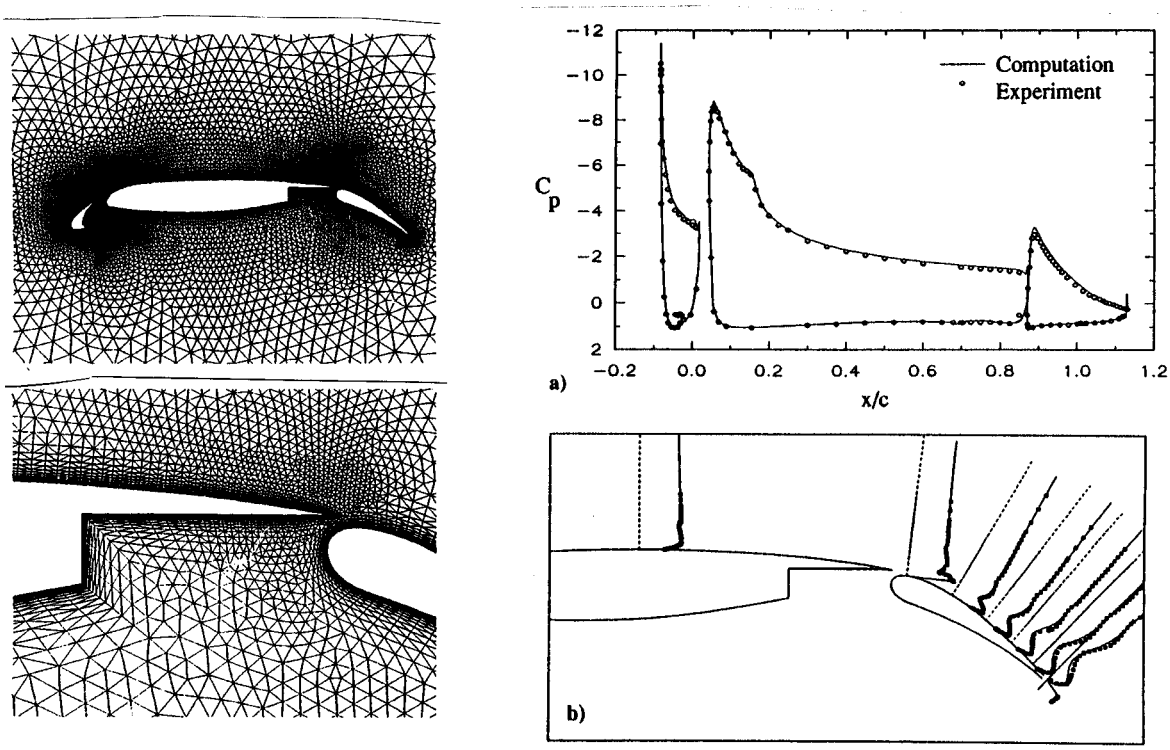


Figure 6. - Unstructured triangular viscous grid and viscous-flow solution for multi-element airfoil at $M_\infty=0.2$, $a=16.2^o$, and $Re_c=9X10^6$: a) surface pressure distributions and b) velocity profiles.

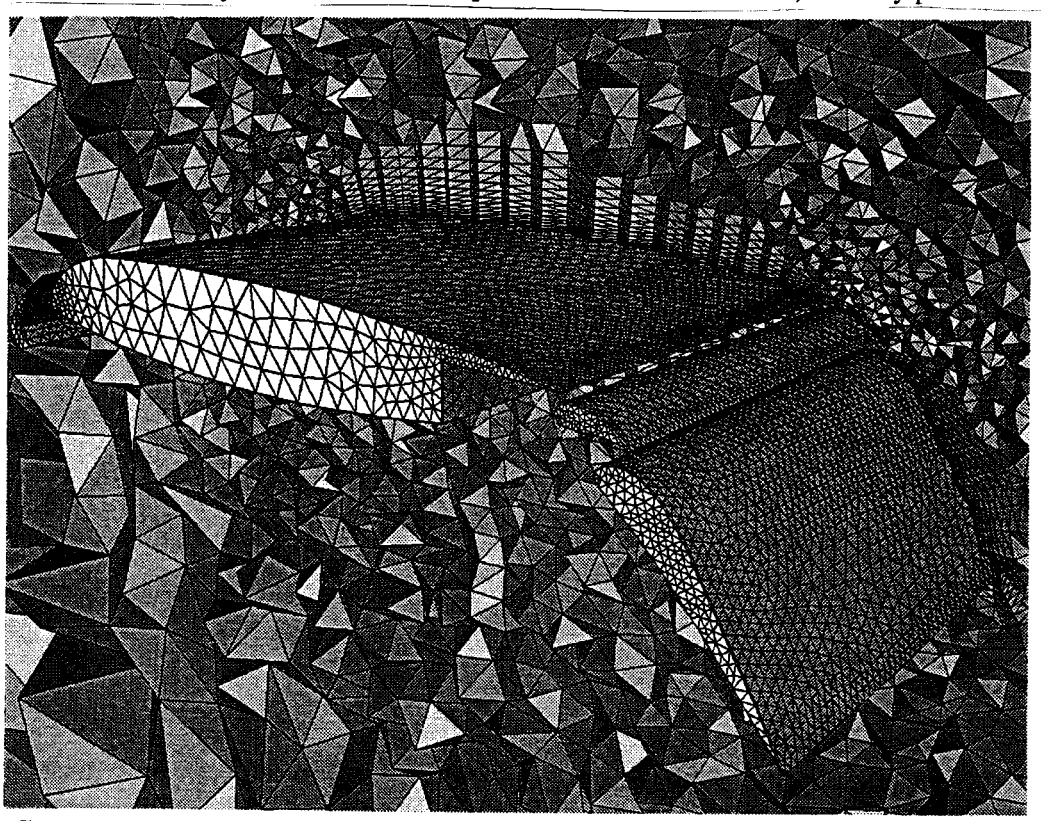
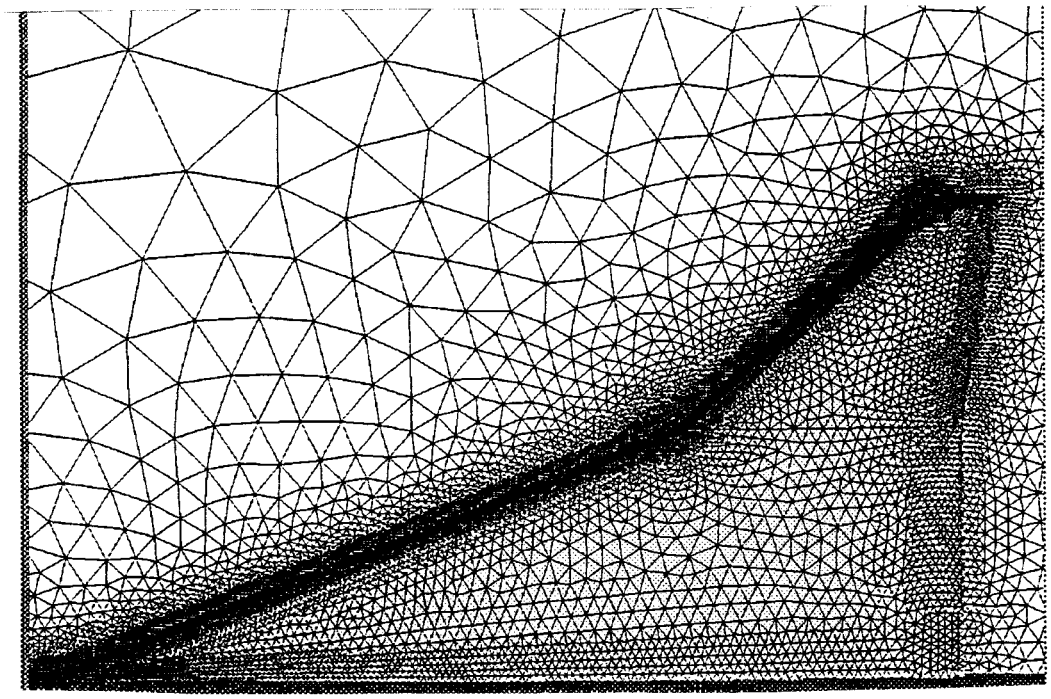
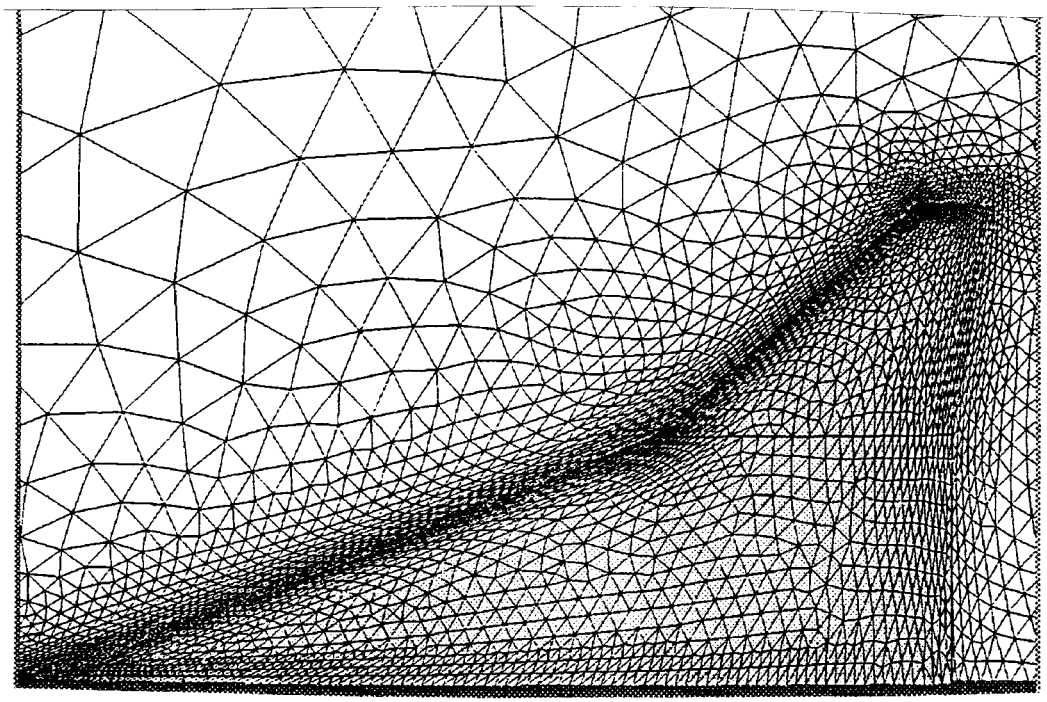


Figure 7. - Complete tetrahedral viscous grid on a multi-element wing shown on a plane.



(a) Isotropic surface and field grid (40,778 triangles on wing).



(b) Multidirectional anisotropically stretched surface and field grid (6,841 triangles on wing).

Figure 8. - Example of multidirectional anisotropic grid stretching for cranked-wing configuration.

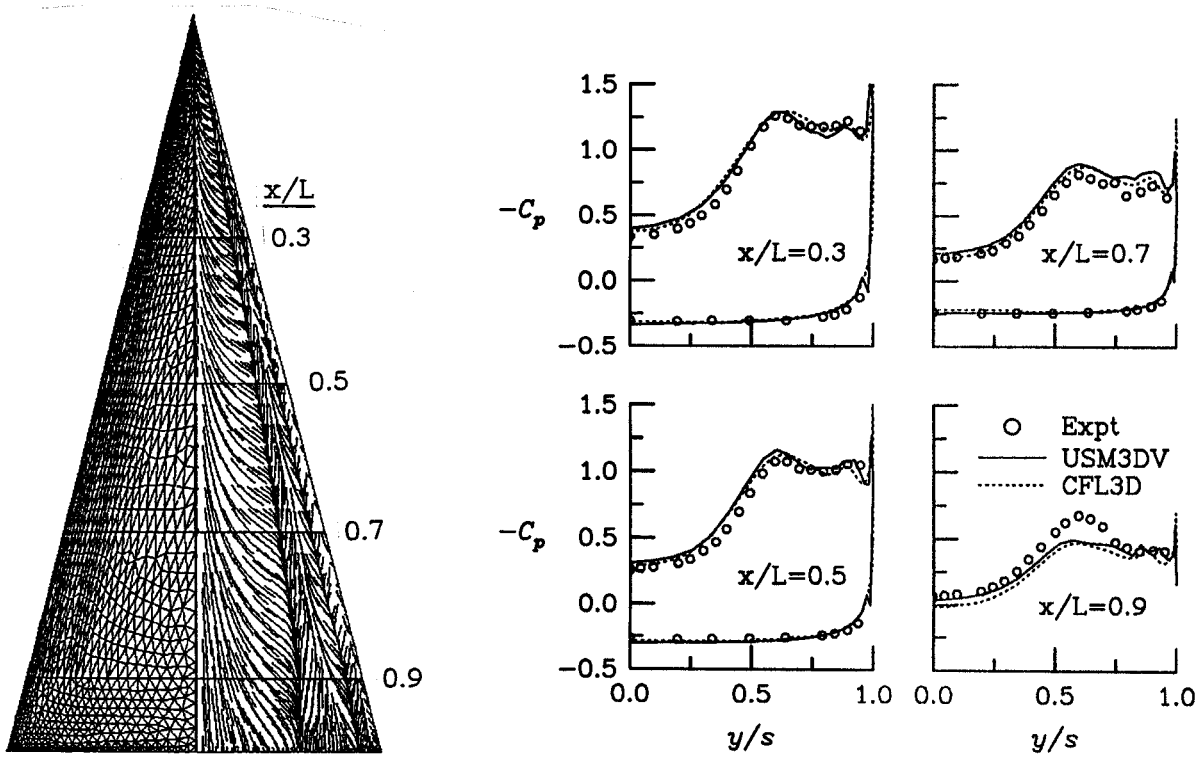


Figure 9. - Laminar flow over aspect-ratio 1 delta wing for $M_\infty=0.3$, $a=20.5^o$, $Re_L=0.9X10^6$. Unstructured surface grid and "oil-flow" pattern (left), pressure profiles at four chord stations (right).

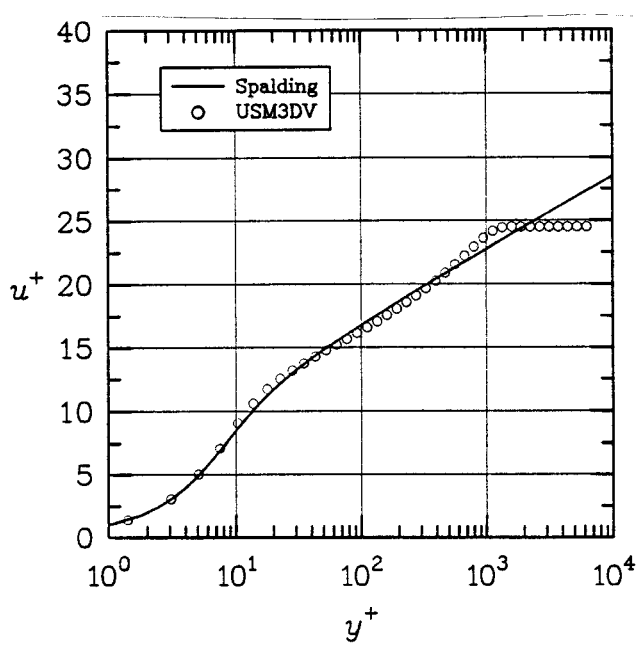


Figure 10. - Comparison of computed law-of-the-wall with analytical model for flat plate boundary layer flow, $M_\infty=0.5$, $Re_L=2X10^6$.

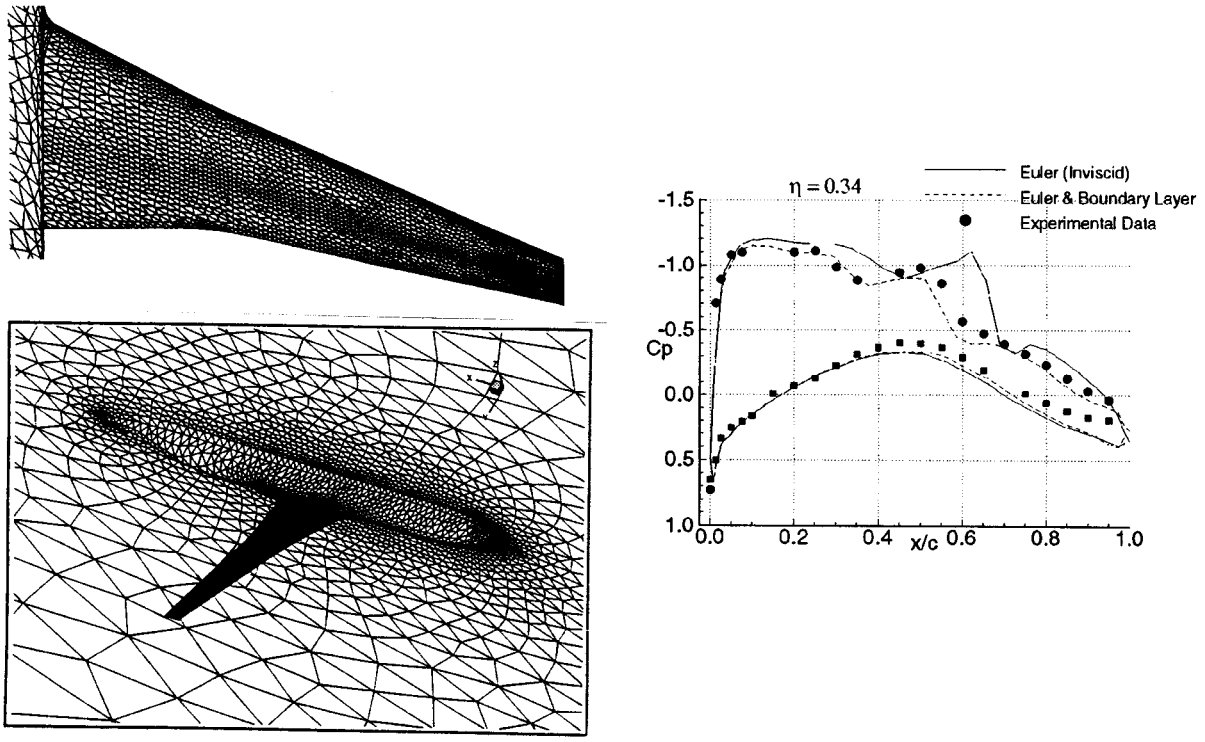


Figure 11. - Effect of interacting boundary layer on pressure predictions for a low-wing transport. M_∞=0.77, α=1.12°, and Re_c=2.6X10^6

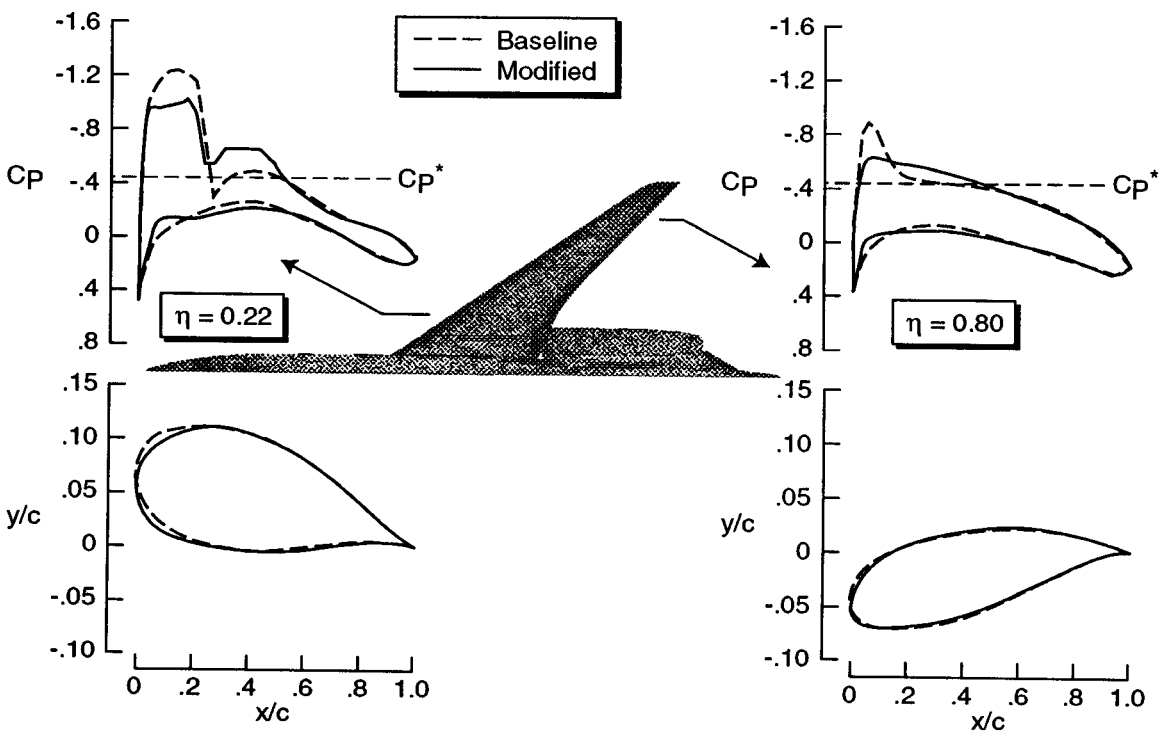


Figure 12. - Application of USM3D/CDISC design method to wing redesign of an executive transport configuration, M_∞=*0.80*.

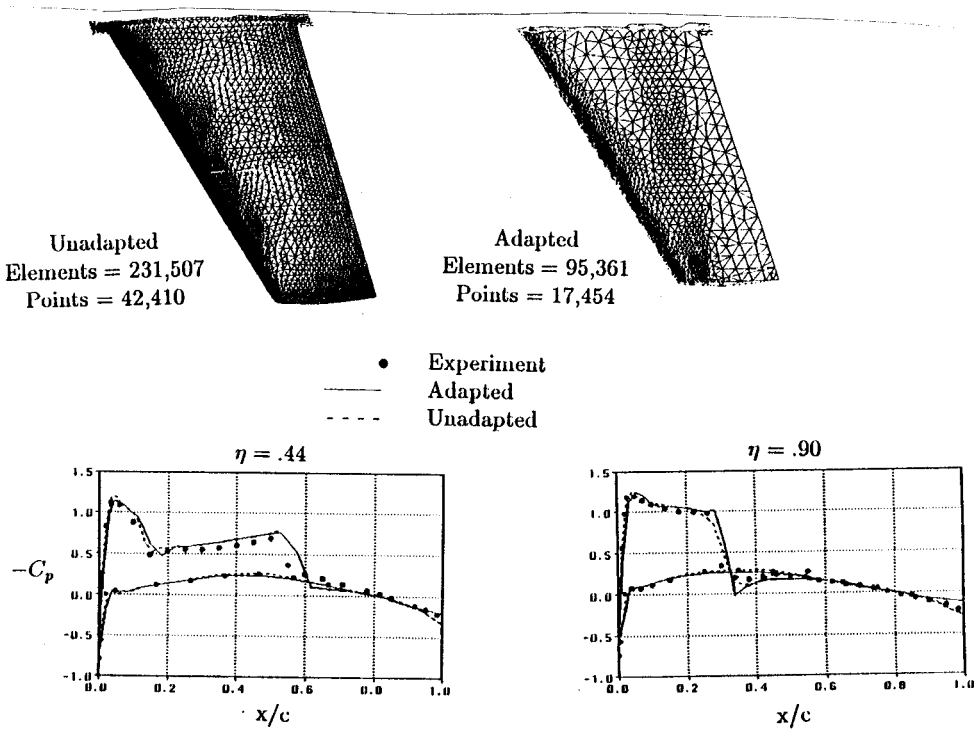


Figure 13. - - Solution derived grid adaptation to shock dominated flow features for ONERA M6 wing, $M_\infty=0.84$, $a=3.06^o$.

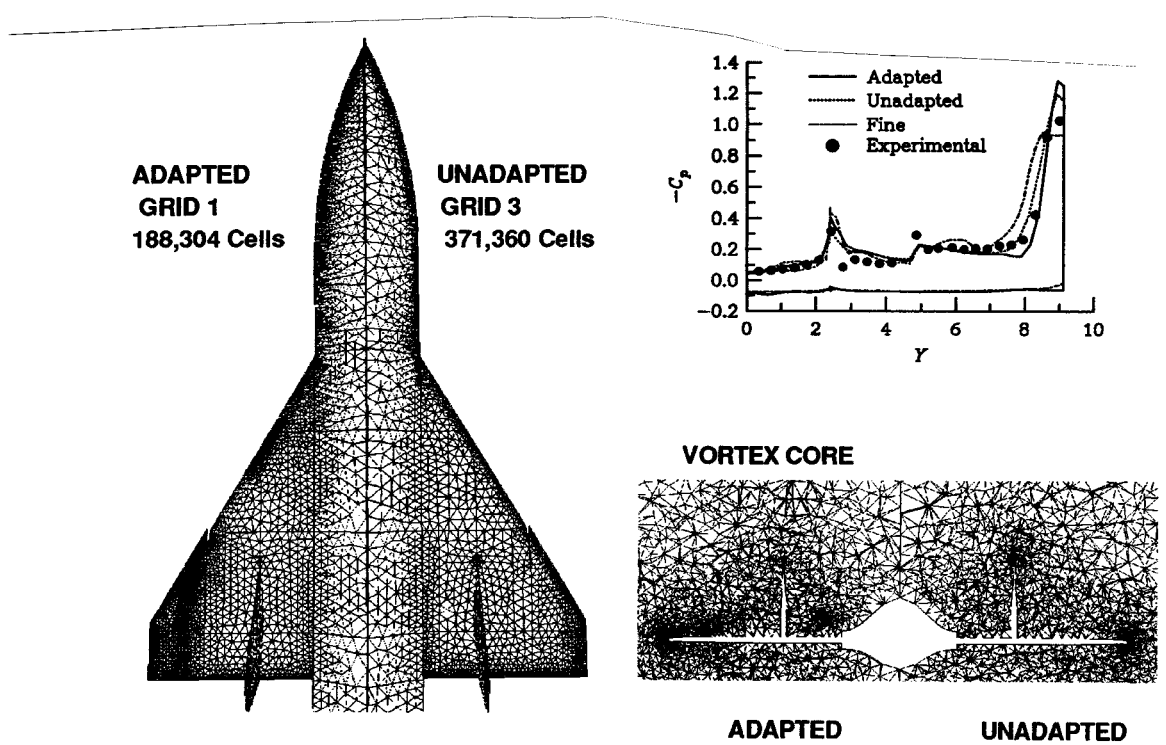


Figure 14. - - Solution derived grid adaptation to vortex-flow features for MTVI configuration, $M_\infty=0.4$, $a=10.5^o$. Surface triangulation (left), field adaptation at $x/c_r=0.93$ (lower right), and surface pressure comparison at $x/c_r=0.93$ (upper right).