N92-24429

BATCH MODE GRID GENERATION: AN ENDANGERED SPECIES?*

David M. Schuster
Lockheed Engineering and Sciences Company
Hampton, VA 23666

SUMMARY

The title poses a rhetorical question which, given the rapid development of interactive surface modeling and grid generation techniques, may appear to have some foundation. In fact, non-interactive grid generation schemes should thrive as emphasis shifts from development of numerical analysis and design methods to application of these tools to real engineering problems. A strong case is presented for the continued development and application of non-interactive geometry modeling methods. Guidelines, strategies and techniques for developing and implementing these tools are presented using current non-interactive grid generation methods as examples. These schemes play an important role in the development of multidisciplinary analysis methods and some of these applications are also discussed.

INTRODUCTION

A great deal of emphasis has recently been placed on the development of interactive software tools for surface modeling and grid generation. This emphasis is motivated and justified by the rapid expansion in capability, and availability of three-dimensional graphics workstations and by the acute need for geometry modeling and grid generation tools capable of accurately describing general, complex configurations. Widely used programs such as GRIDGEN[1], EAGLE[2], and more recently, EAGLEView, have addressed these issues.

The availability of these interactive geometry modeling systems has made it convenient for applications researchers, who previously relied on outside sources for pre-generated grids, to generate their own grids. Thus, these researchers progressed from performing virtually no grid generation to using tools designed to model general, complex geometries using general grid topologies. Often overlooked in this progression is a large database of research and software development for grid generation schemes which operate in a non-interactive or batch mode.

Current interactive schemes provide an outstanding resource for modeling complex multicomponent geometries using arbitrary grid topologies. In many cases, they provide the only means by which some geometries, like engine inlet/airframe intersections, single and multiple stores, etc., can be accurately modeled. However, for a great number of problems, this level of modeling detail and flexibility is not required. Therefore, many researchers new to the field of grid generation needlessly spend countless man-hours developing surface geometry databases, distributing and redistributing points, and generating

* Work performed under NASA Contract NAS1-19000

surface, and ultimately, field grids using interactive programs to model relatively simple geometries.

This paper focuses on schemes designed to efficiently generate grids in one step, starting with a simple definition of the surface geometry of the configuration to be modeled. Specifically, it discusses the techniques and strategies used to develop the Complete Aircraft Mesh Program (CAMP)[3]. CAMP is a multi-block grid generation program which was originally developed to model complete fighter aircraft configurations including wing, fuselage, canard, horizontal stabilizer and symmetry plane mounted vertical fins. It has been extended to model aircraft with vertical fins mounted away from the symmetry plane. Grids have been generated for a wide range of fighter aircraft, transports, and most recently, a generic National Aerospace Plane (NASP) configuration. Examples of grids generated by CAMP and programs developed using various CAMP modules will be discussed to demonstrate the utility of non-interactive grid generation methods for modeling of relevant engineering problems.

The paper also addresses the importance of non-interactive grid generation methods in analyzing multidiscipline engineering problems. Unsteady aerodynamics, aeroelasticity and aeroservoelasticity are examples of disciplines that require deforming geometries to be efficiently modeled while simultaneously calculating the flowfield properties and aerodynamic loads. The strategies used to develop grid generation programs such as CAMP can be efficiently employed within an aerodynamic analysis program to dynamically deform the surface and field grids according to a specified or calculated motion. Examples of how these schemes have been implemented in a three-dimensional aeroelastic analysis method employing Euler/Navier-Stokes aerodynamics will be presented to amplify this point.

The discussion presented in this paper is applicable to the development of single and multiple zone structured grids. Unstructured grid generation presents a completely different set of challenges. The discussion is also primarily directed toward the applications researcher, rather than the algorithm developer. Applications researchers are typically required to model a wide variety of unique geometries and are usually the first to invalidate a given grid generation method. Thus, they typically are in the greatest need of new and more efficient ways to accomplish their work.

STRATEGIES AND TECHNIQUES USED IN THE DEVELOPMENT OF THE COMPLETE AIRCRAFT MESH PROGRAM (CAMP)

CAMP was originally written to provide numerical modeling support for a Euler/Navier-Stokes Three-Dimensional Aeroelastic (ENS3DAE) analysis program developed by the Georgia Division of the Lockheed Aeronautical Systems Company (LASC-GA) under Air Force Contract F33615–87–C-3209, "Flight Loads Prediction Methods for Fighter Aircraft." The main objective of developing the program was to provide a capability which does not require a large amount of user interaction or a high level of grid generation expertise to model complete, structurally flexible, fighter aircraft. The approach taken was to develop a single grid generation program which uses a fixed grid topology that is applicable to as wide a range of geometries as possible. Among the requirements for this development were that the program be made highly modular in structure, use a simple but general input structure and execute efficiently on a wide range of computer hardware. The result of this effort is a program which has been used to model geometries as simple as an isolated wing to complex wing/fuselage/horizontal tail/vertical tail configurations. CAMP has been run on platforms ranging from supercomputers to low-level workstations.

CAMP exploits the inherently modular nature of aircraft geometries to obtain its high degree of efficiency and flexibility. The majority of fixed wing aircraft can be represented by one to ten major components. CAMP can currently model up to six major components: wing, fuselage, canard, horizontal tail and one or two vertical tails. The input data is arranged according to these major components using a combination of Namelist and free-format input as shown in Figure 1. The geometry input was designed to be as intuitive as possible so that surface data could be obtained from a designer using a Computer Aided Design (CAD) system and converted to the CAMP format with minimal effort. Wing, horizontal tail and canard geometries are input as a series of constant spanwise station cuts, fuselage data as constant axial station cuts and vertical tail data as constant vertical station cuts. Individual components can be added or deleted simply by changing a binary switch in the Grid Control and Index Specification input. Since the program was written primarily from an aerodynamicist's perspective, it assumes that all configurations have a wing, and this component is a required input for every configuration. However, through creative manipulation of the input data, fuselage alone and most fuselage/tail (missile) geometries can be readily modeled using CAMP.

CAMP Input Processor

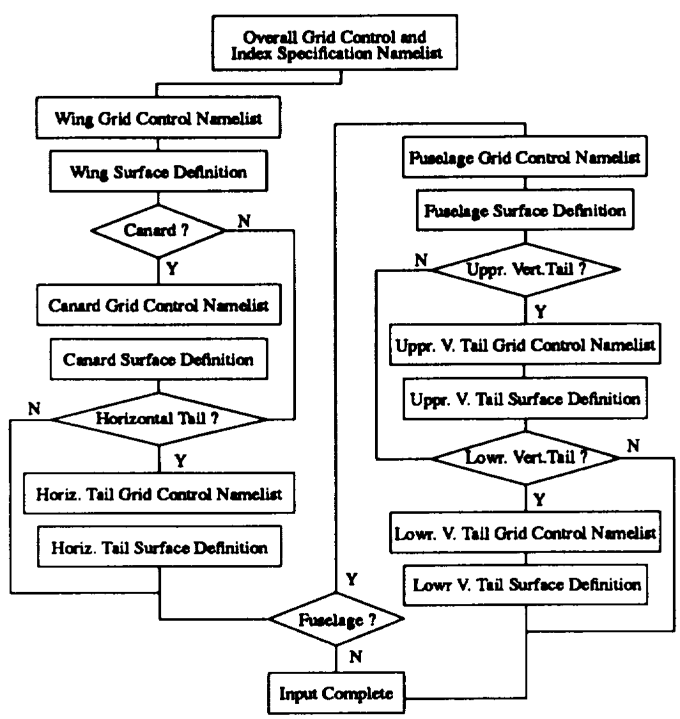

Figure 1. Modular input structure of CAMP.

Previous research indicated that a zonal H-H grid topology, which has the desirable property that the resulting grids are Cartesian like and closely aligned with the streamlines of the flow, could be used to accurately model a wide range of geometries and is especially suited to fighter and transport aircraft[4,5,6]. Thus, a zonal H-H grid topology, shown in Figure 2, was adopted for use in the program. At present, a given grid is divided into a minimum of two and a maximum of six zones or blocks. All blocking is performed automatically within the code depending on the combination of components chosen to be modeled. Geometries consisting of combinations of wing, canard and/or horizontal tail only are modeled using two blocks, one for the upper section of the flowfield, and one for the lower section of the flowfield. Addition of a fuselage automatically ensures that at least two more blocks will be added, one for the region above the fuselage and one for the region below the fuselage. Symmetry plane mounted vertical tails can be added without additional blocks, but offset vertical tails, as encountered on many modern fighters, require that one or both of the fuselage blocks be split into two blocks.

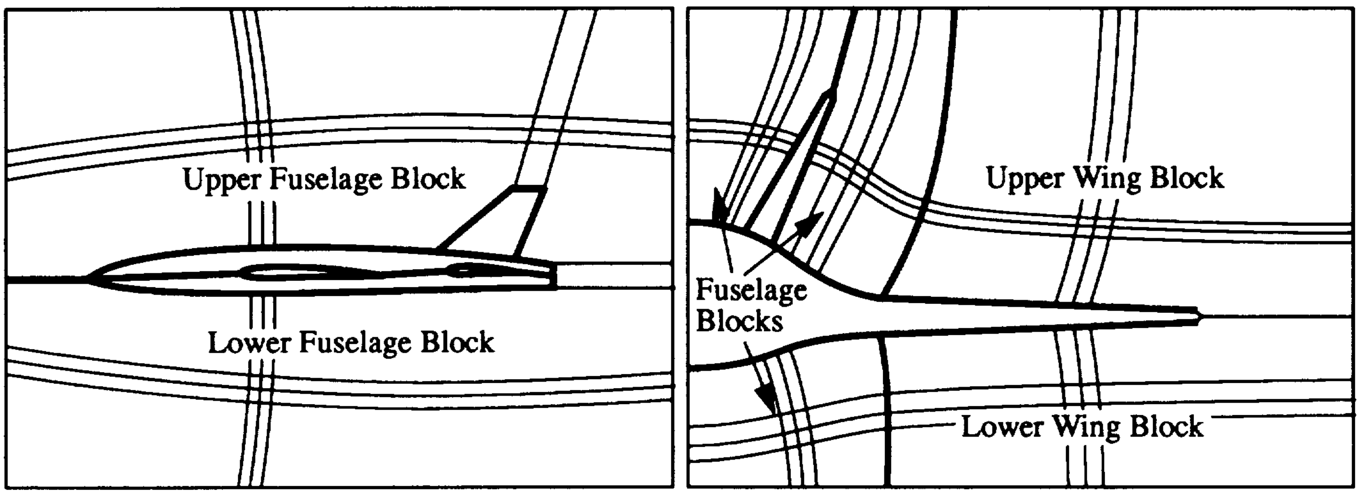

Figure 2. CAMP zonal H-grid topology.

The complete three-dimensional grid is generated in one, two, or three steps, depending on the complexity of the configuration. Like the input processor, the surface and field grid generation also proceeds in a modular fashion. The zones about the horizontal surfaces (wing, canard and horizontal tail) are always generated first. If a fuselage is to be modeled, its grid zones are generated next using the previously defined horizontal surface zones to guide the fuselage grid generation. Finally, the addition of vertical tail surfaces requires manipulation of the fuselage grid, which consists of a series of algebraic shearings and redistribution of points along existing grid lines. The grid generation can be stopped after any of the above steps, and a fully defined three-dimensional grid of the components processed to that point can be output. Therefore, generating grids about different combinations of components entails nothing more than toggling software switches in the input data.

CAMP generates all surface grids using algebraic techniques. The organization of the input data allows the surface grids to be defined using a series of simple, one-dimensional parametric interpolations. For instance, the wing surface grid is generated by first interpolating in the streamwise direction using a set of input grid point indices and spacings. This results in a uniform streamwise distribution of grid points at each defining station. This streamwise interpolated surface is then interpolated in the spanwise direction to obtain the complete surface grid. A similar procedure is performed for each of the remaining components.

An important feature of the CAMP surface grid generation scheme is that all interpolations are linear. Ironically, this is done to preserve the accuracy of the input surface data. Many realistic aircraft configurations employ sharp edges, such as supersonic wing leading edges, forebody strakes and chines, etc., which play an important role in the aerodynamics of the vehicle. Use of higher order interpolation methods can result in spurious oscillations of the surface data near sharp edges. Some splines, such as tension, Bezier and B-splines, can be used to interpolate without oscillation near these points, but these either require additional user input to control the interpolation, or inordinately smooth the sharp edge. Linear interpolation guarantees that a sharp edge will be captured as long as there is a point on the edge. By using linear interpolation, CAMP shifts the burden of an accurate surface definition to the input data which is presumably coming from a CAD package or an analytical definition. It is generally a trivial exercise to vary the density of points describing the surface using these sources, and thus it is easier to control the accuracy of the surface definition external to the grid generation program than internally.

Field, or volume, grids are generated using a combination of algebraic and elliptic equation techniques. The fields above and below the horizontal surfaces are generated using a quasi-three-dimensional method where two-dimensional grids are defined at each spanwise station and stacked together to define the complete grid. The individual planar grids can be generated using an algebraic, parabolic[7], or fully elliptic[8,9] scheme. The algebraic scheme is always used to generate an initial grid for the parabolic and elliptic methods. It simply performs a linear interpolation between the wing/canard/horizontal tail surface and the outer boundary, with a specified clustering near the horizontal surfaces. The parabolic scheme combines an algebraic and hyperbolic equation grid generation scheme to march from the horizontal surface to the outer boundary. Along the way, it uses an elliptic equation grid generator to smooth the grid. The parabolic scheme has the desirable feature that near the initial horizontal surface, the grid lines transverse to the surface are nearly orthogonal and clustering is easily controlled. However, for geometries with very blunt leading edges, the parabolic grid generator can produce very artistic, but unusable, grids. To counteract this problem, an elliptic grid generator has been added as a post-processor to the parabolic scheme to further smooth the grid. If a large number of post processing iterations are specified, the entire grid generation scheme is equivalent to generating the grid planes using only the elliptic equation solver.

The field grids above and below the fuselage are generated using purely algebraic methods. The fuselage surface grid is comprised of a series of constant axial station cuts whose positions are defined by the axial locations of the grid points in the previously defined horizontal surface grid at its intersection with the fuselage. A series of algebraic shearings, linear interpolations and smoothings are then used to generate the field grid between each fuselage station cut and the outer boundary. Vertical tail surfaces are added by shearing and, where necessary, re-interpolating the fuselage grid. The horizontal surface zones are mated to the fuselage zones by algebraically shearing the wing grid in the spanwise direction.

At present, there is point to point continuity at the zonal interfaces, but the slopes and transverse grid spacings across the interfaces are discontinuous. The current aerodynamic analysis program supported by this grid generator does not require slope and spacing continuity across the interfaces since it does not use differences across the zones. Also, there is no full three-dimensional elliptic equation grid generation capability in the program. If grids with smooth zonal interfaces are required, CAMP could be easily modified to output its grid coordinates and zonal interface information in a format suitable for input into the 3–D elliptic grid generators associated with the GRIDGEN or EAGLE programs. In effect, CAMP would become a preprocessor for these codes. A three-dimensional grid generation module could also be added to CAMP with minimal effort. Surface grids generated by CAMP have also been used to define surface databases for the GRIDGEN software.

EXAMPLES OF GRIDS GENERATED BY CAMP AND ITS DERIVATIVES

Numerous aircraft configurations have been modeled using CAMP. These geometries range from isolated wings to wing/fuselage/tail configurations. A generic fighter configuration, which possesses many of the features present on modern aircraft designs, will be used to demonstrate some of the capabilities of the program. The geometry consists of a wing, fuselage and twin, canted vertical tails, as well as a fuselage forebody chine. The surface grid generated by CAMP for this geometry is shown in Figure 3. The half-span surface of the configuration is modeled using a total of 13,275 points, 2,394 on the wing, 1,806 on the vertical tail and 9,075 on the fuselage. The full-span geometry shown in the figure is obtained by reflecting the half-span grid about the fuselage symmetry plane.

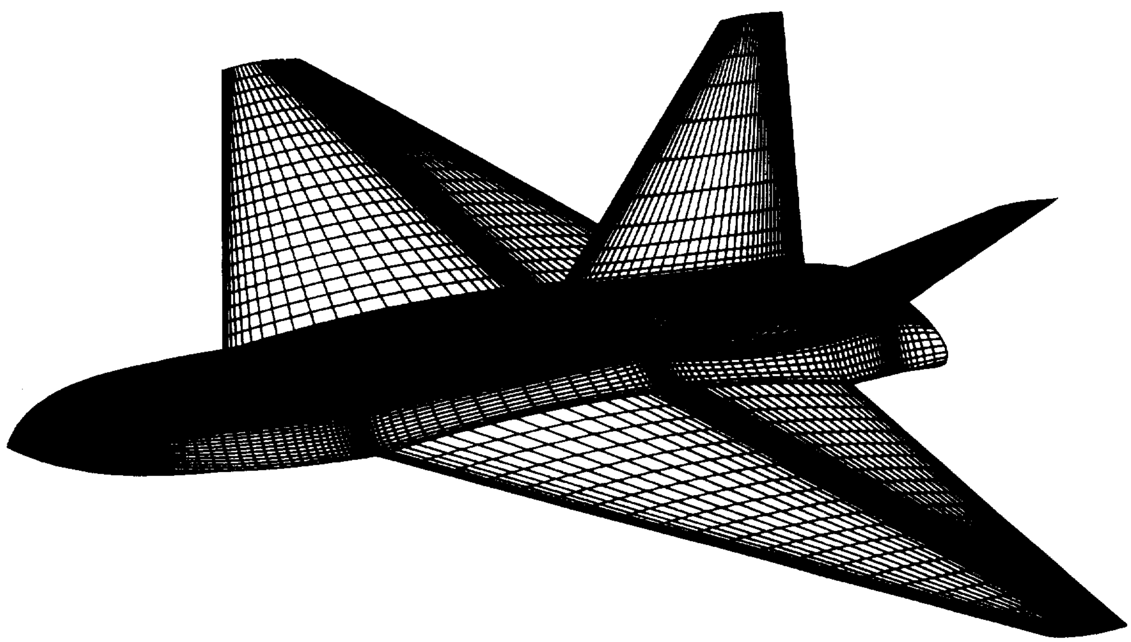

Figure 3. CAMP surface grid for a generic fighter configuration.

CAMP generates a total of five field grid zones for this configuration, one each modeling the flowfield about the upper wing, lower wing, and lower fuselage surfaces. The upper fuselage surface is broken into two zones, one inboard of the vertical tail and one outboard. The grid generated for this case has a total of 693,816 grid points for the half-span model. Each zone consists of 141 points in the streamwise direction and 37 points in the normal direction. All of the zones have 29 points in the spanwise direction except for the zone between the vertical tail and the wing grid which only has 17 spanwise points. All of the aircraft features are accurately modeled by this grid topology and point distribution.

Figure 4 shows a side view of a portion of the grid generated by CAMP at the wing mid-station. The figure shows the grids obtained using each of the three available grid generation schemes. The algebraic grid consists of straight lines connecting the wing surface to the outer boundary , and the streamwise lines are clustered near the surface to capture the wing boundary layer. The parabolically generated grid maintains a high degree of orthogonality at the wing surface and along the wake lines ahead of and behind the wing. The elliptic grid trades surface orthogonality for a higher degree of smoothness in the interior grid.

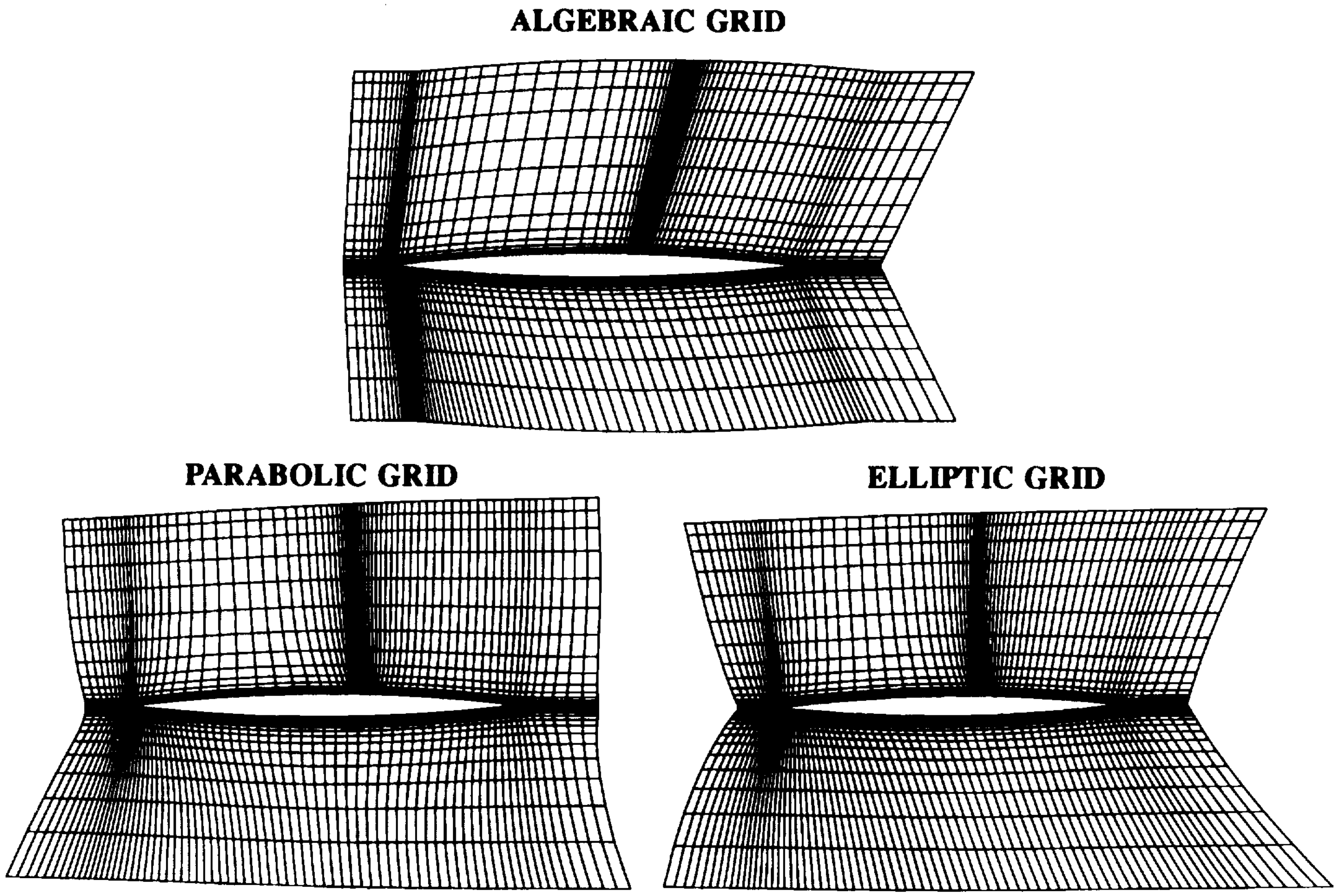

Figure 4. Side view of H-H grid generated about the wing midspan.

A side view of the fuselage grid at the plane which contains the vertical tail is shown in Figure 5. The vertical tail planform is highlighted by the dense clustering of lines near the aft end of the fuselage. The aircraft is modeled as if it were mounted on a sting, but the fuselage could also be closed off or another grid zone added in the sting area to simulate the engine exhaust plume. This grid is generated algebraically, and it accurately defines all areas where high flow gradients are anticipated, including the fuselage nose and boundary layer and the wing and vertical tail leading and trailing edges.

A front view of a section of the grid around the fuselage, vertical tail and wing is shown in Figure 6. In this figure, the interfaces between the various zones are clearly evident. The figure shows the discontinuity of slope and spacing across the zonal interfaces, but the grids within each zone are well behaved.

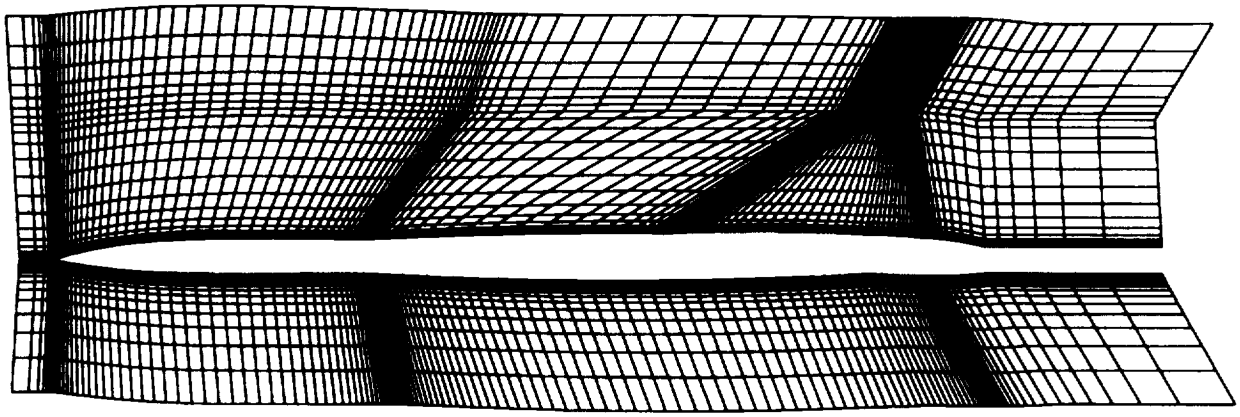

Figure 5. View of the algebraic fuselage grid through the vertical tail plane.

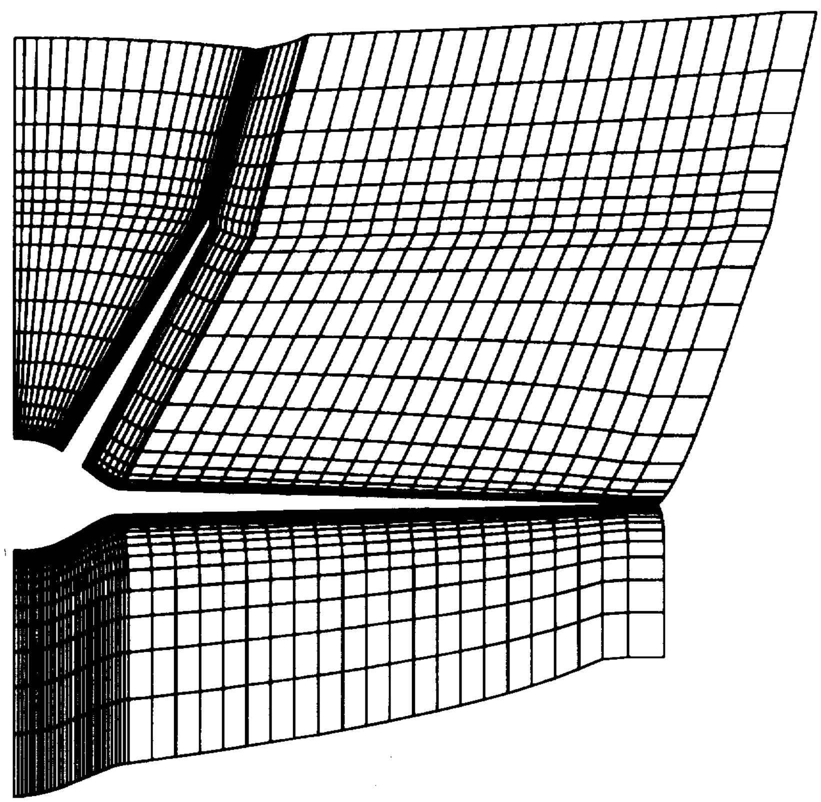

Figure 6. View of patched, zonal grid through the fuselage, vertical tail and wing.

The Cray 2 (Voyager) supercomputer at the NASA Langley Research Center was used to generate these examples. On this computer a grid suitable for aerodynamic computations was generated in less than 15 CPU (Central Processor Unit) seconds. Generation of the grid using the algebraic and parabolic methods are significantly more efficient than the elliptic equation scheme, with the algebraic grid requiring approximately 4.5 seconds and the parabolic grid requiring 14.2 seconds. The elliptic equation grid was defined using 50 iterations at each wing station and required 99 seconds. These times represent a complete grid generation, including geometry input, surface grid, outer boundary and zonal interface specification and final generation of the full 3–D grid, demonstrating CAMP's high degree of computational efficiency.

It is not the intention of this paper to promote CAMP as the ultimate grid generation program for every application. Therefore, this section will be closed by presenting two examples of grids generated by other non-interactive grid generation programs which are based on the above grid generation strategies. These examples are used to illustrate a number of important points pertaining to the utility of non-interactive grid generation methods. Most importantly, the notion that adoption of a given non-interactive grid generation scheme forces the researcher to adhere to a specific grid topology is dispelled. The modular coding employed in the development of CAMP has been used to advantage in the development of a number of other non-interactive grid generation programs using other than an H-H topology. Both of the following configurations were modeled with single zone C-H grid topologies using a program based on many of the modules in CAMP. Some of these modules, such as the point distribution and clustering routines and the algebraic and elliptic equation grid generators were implemented as "black boxes" with virtually no changes in the original coding. Other routines required minor modifications to be suited to the new grid topology. The input formats for the definition of the various aerodynamic surfaces are identical to those used in CAMP.

Figure 7 shows the surface grid for a generic fighter wing/body developed using the above C-H topology grid generator. The geometry consists of many of the overall features associated with modern fighter aircraft, including a blended wing/fuselage and a highly swept fuselage forebody. This geometry could be easily modeled using CAMP, but the wing and fuselage are so highly blended that a single zone grid models this configuration much more efficiently than CAMP's multi-zone topology. In this case, the entire wing/fuselage configuration is modeled as a wing alone. This allows the geometry and its associated flowfield to be accurately described using a minimum number of grid points without compromising the salient geometric features of the model. Selection of this modeling technique reduced the grid generation effort for this vehicle to a relatively simple exercise and allowed the researchers to focus their attention on accurately predicting the complex flow physics associated with the operation of this type of aircraft at high angles of attack[10].

The applications researcher is often required to perform parametric studies involving a number of geometric changes to a given configuration. These may be as simple as minor changes to airfoil contours on a wing to more complex perturbations like control surface deflections. For these types of problems, each analysis requires that a new grid be generated. If an interactive grid generation method is used, the manpower required to model the various geometries can be formidable. Application of non-interactive methods can greatly relieve this effort. An example of such an application is presented in Figure 8. This surface grid represents a wing with multiple, deflected, leading and trailing edge control surfaces. The C-H grid generation program described above was used to generate the grid about this wing by modifying the input to allow the user to specify an arbitrary number of leading and trailing edge control surfaces and their respective deflections. Thus, by changing a handful of parameters in the program input, the researcher can efficiently generate a series of grids with various control surface sizes and deflections. This discussion leads to the final topic of this paper which addresses multidisciplinary engineering problems.

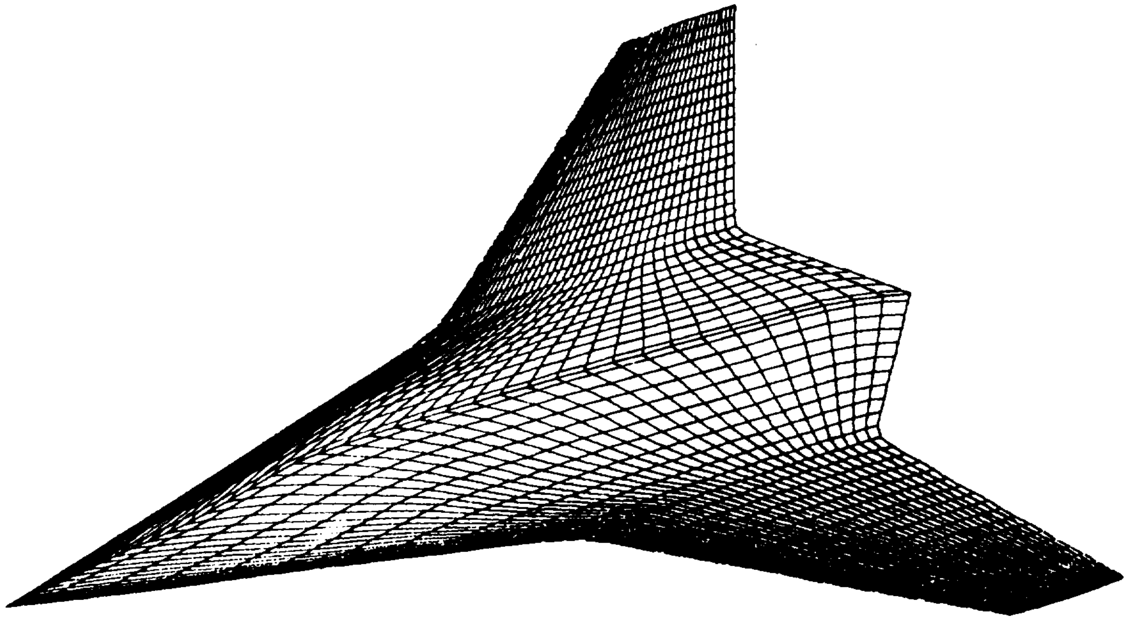

Figure 7. Surface grid for a blended wing/body fighter geometry.

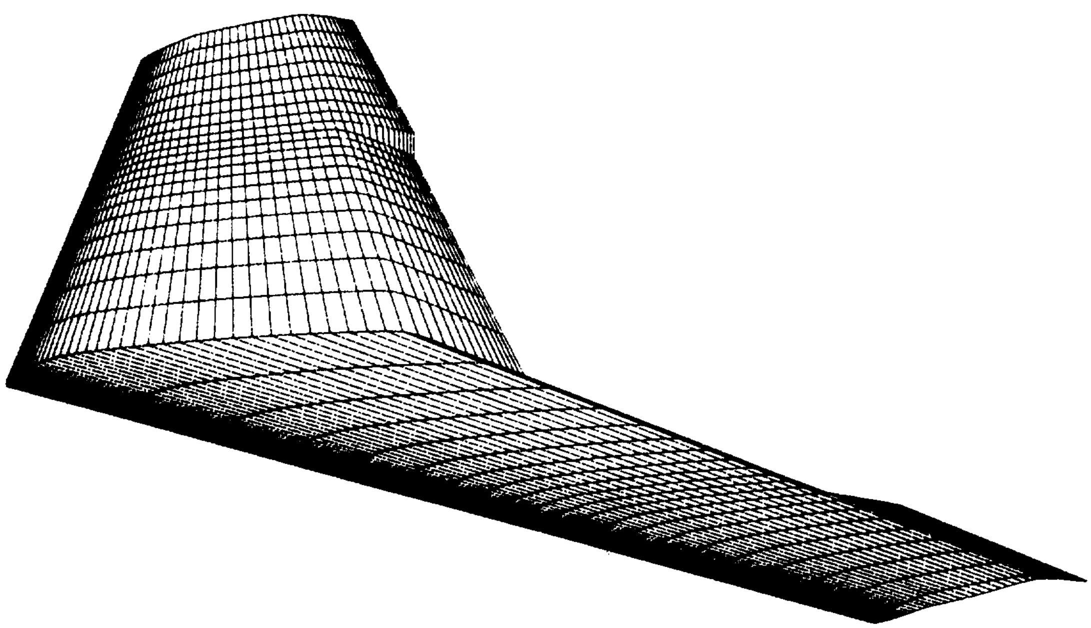

Figure 8. Surface grid for a wing with deflected leading and trailing edge control surfaces.

GRID GENERATION FOR MULTIDISCIPLINARY ENGINEERING PROBLEMS

The problems associated with grid generation for parametric analyses are elevated to an even higher level when one examines the modeling requirements of multidisciplinary engineering analyses. For these problems, the geometry is required to "react" to forces produced by the interaction of phenomena that are peculiar to several engineering disciplines. For instance, aeroelasticity deals with the interaction of the aerodynamics with the structure of a flexible vehicle. In this case, as the aerodynamic load changes, the structure deflects, which in turn changes the aerodynamic load. This process continues until a steady state load and structural deflection are achieved, or in many cases the cycle can continue indefinitely. Aeroservoelasticity is even more complex with aerodynamics, structures and controls all interacting. In each of these cases, the geometry of the vehicle changes, often thousands of times, throughout the course of the analysis. It is not practical to stop the calculation after each geometry perturbation to externally generate a new grid. Therefore, a grid generation capability must be included as an integral part of the multidisciplinary analysis method.

Until recently, researchers have used approximate methods to describe geometric deflections for these types of problems. Many of these involved perturbing the velocity boundary condition at the vehicle surface to simulate the deflection of the geometry. For small deflections, these methods were satisfactory, but many problems preclude the use of these methods. More recent developments have concentrated on physically deflecting the vehicle surface and likewise, the grid surrounding the vehicle. The ENS3DAE aeroelastic method uses a computationally efficient, robust algebraic redistribution algorithm, based on methods developed for CAMP, to update the grid surface geometry and field grids iteratively during an aeroelastic analysis. Batina[11] has employed a spring analogy which has physical basis and is easily applied to unstructured grids as well as structured grids. Kandil and Chuang[12] solve an unsteady equation, known as the Navier-Displacement equation, for the grid motion. Methods employing an elliptic equation grid generator for a small number of iterations to smooth the field grid after each perturbation of the surface geometry have also been suggested. Each method has its relative merits and shortcomings, typically involving a trade-off between computational efficiency and general applicability. Due to its relative simplicity and robust operation, the following paragraphs discuss the method used in the ENS3DAE aeroelastic analysis method. However, while this method is applicable to a wide range of problems, this area is experiencing a high degree of research activity, and the reader is encouraged to examine alternative methods as they are published.

The method employed in ENS3DAE deflects the grid using a simple algebraic shearing which preserves the initial quality of the grid. Originally, the grid was assumed to deflect only in the vertical (Y) direction, but the method has since been extended to accept deflections in all three directions. Figure 9 will be used to describe the original implementation of the algorithm. The basic concept is to update the interior grid such that points near the surface move with the surface, while points far from the surface do not move significantly. This allows grids initially clustered near the surface to resolve viscous flow phenomena, to remain clustered at the surface, while the outer boundary of the grid remains fixed in space. This is accomplished by computing a normalized arc-length distribution for each grid line connecting the deforming surface to the outer boundary. The deflection of each grid point along that line is then computed by:

$$\Delta Y_K = \Delta Y_{AE}\left(1 - S_K/S_{MAX}\right) \tag{1}$$

where, ΔY_{AE} is the deflection at the surface, S_K is the arc-length along the grid line at point K and S_{MAX} is the total arc-length of the grid line. While this is a very simple and computationally efficient method for updating the grid, it is also quite effective, as illustrated in Figure 10. This figure shows a front view of a CAMP generated wing/body grid before and after an aeroelastic calculation using ENS3DAE. The deflection for this case is significant, and the overall features of the original grid, such as the clustering of grid lines near the vehicle surface is well preserved in the deflected grid. This grid was updated a total of 1000 times during the aeroelastic calculation with no apparent loss of resolution. The time required to perform the grid updates is negligible compared with that required to solve the flow equations and the structural equations of motion.

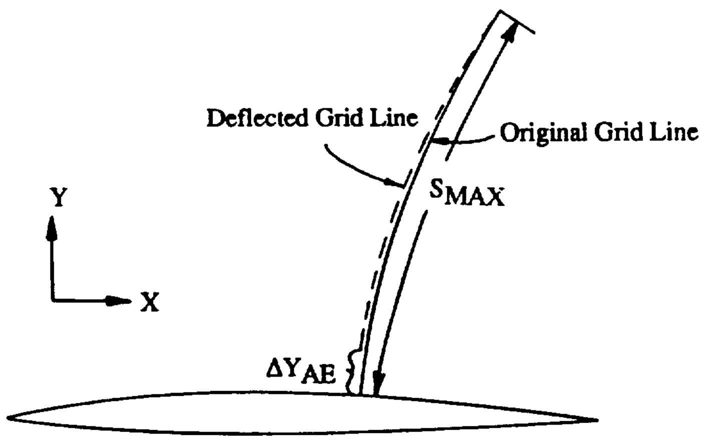

Figure 9. Algebraic grid redistribution method.

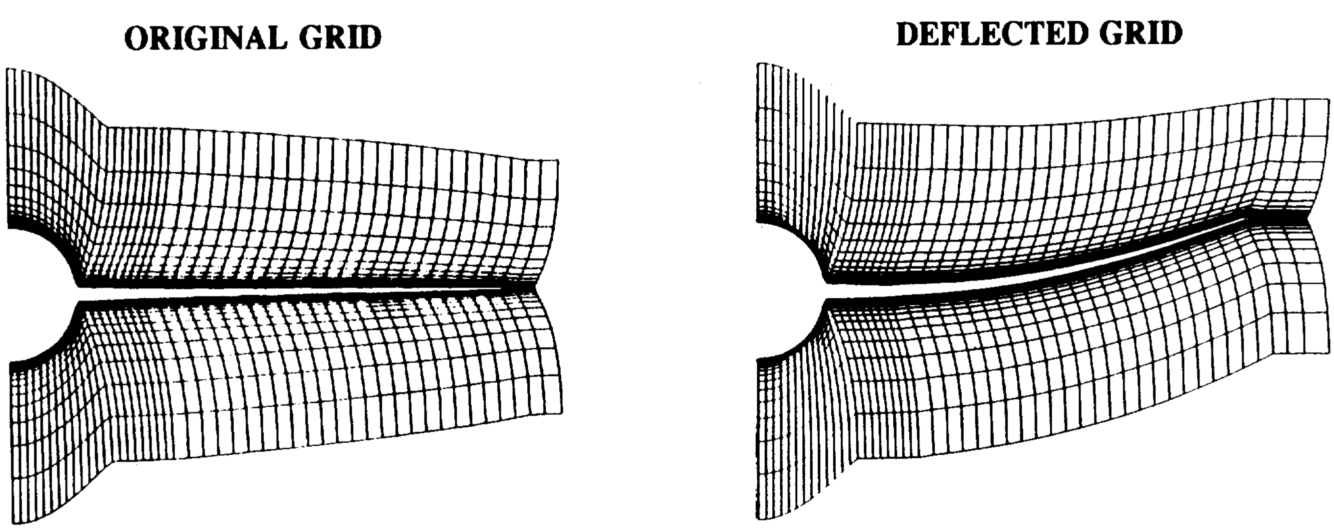

Figure 10. Original and deflected wing/body grid.

CONCLUSION

This paper has discussed the relative merits of geometry modeling and grid generation using methods based on a non-interactive, or batch, mode of program execution. Efficiency and improved productivity have been the primary motivation used to develop an existing non-interactive grid generation program known as CAMP. The overall features of the program, as well as its performance and examples of grids generated by the method have been presented. Methods for generating grids with alternative topologies, and implementation of some of these techniques into multidisciplinary engineering problems has been introduced.

The purpose of this paper is not to "sell" the reader on the CAMP grid generator or any other specific method, but rather to introduce techniques and concepts used to develop this and similar programs. The time and effort which must be devoted to geometry modeling and grid generation for complex configurations has been well documented in literature, and methods utilizing three-dimensional, interactive graphics are well positioned to greatly relieve this constraint. However, there are many problems, some involving relatively complex configurations, which can be effectively modeled using simple, efficient, non-interactive grid generation techniques. Application of these schemes can often reduce the burden of geometry modeling, and significantly improve productivity.

REFERENCES

1. Steinbrenner, J. P., J. R. Chawner and C. L. Fouts, "The GRIDGEN 3D Multiple Block Grid Generation System," WRDC-TR-90-3022, July, 1990.
2. Lijewski, L. E. , et al., "Program EAGLE User's Manual," AFATL-TR-88-117, September 1988.
3. Schuster, D. M. and E. H. Atta, "Flight Loads Prediction Methods for Fighter Aircraft, Volume II — Complete Aircraft Mesh program (CAMP) User's Guide," WRDC-TR-89-3104, November, 1989.
4. Atta, E., L. Birckelbaw, and K. Hall, "Zonal Grid Generation Method for Complex Configurations, " AIAA Paper 87-0276, 1987.
5. Birckelbaw, L. D., E. H. Atta, and C. S. Rubertus, "Euler Analysis and Correlations for Realistic Upper Surface Blowing Aircraft Configurations," AIAA Paper 87-2271, August, 1987.
6. Narain, J. P., "Inviscid Flow Predictions for Complex Aircraft Configurations Using Euler Equations," AIAA Paper 87-2616, August, 1987.
7. Noack, R. W. , and D. A. Anderson, "Solution Adaptive Grid Generation Using Parabolic Partial Differential Equations," AIAA Paper 88-0315, 1988.
8. Thompson, J. F., F. C. Thames, and C. W. Mastin, "Automatic Generation of Body-Fitted Curvilinear Coordinate Systems for Fields Containing Any Number of Arbitrary Two-Dimensional Bodies," *Journal of Computational Physics*, Vol. 15, 1974, pp.299-314.
9. Thomas, P. D., "Composite Three-Dimensional Grids Generated by Elliptic Systems," AIAA Paper 81-0996, 1981.
10. Vadyak, J. and D. M. Schuster, "Navier-Stokes Simulations of Burst Vortex Flowfields for Fighter Aircraft at High Incidence," *Journal of Aircraft*, Volume 28, Number 10, October, 1991, pp. 638-645

11. Batina, J. T., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex-Aircraft Aerodynamic Analysis," AIAA Journal, Volume 29, Number 3, March 1991, pp. 327–333.

12. Kandil, O. A. and H. A. Chuang, "Dynamic Grid Deformation Using Navier-Displacement Equation for Deforming Wings," AIAA-89–1982–CP, AIAA 9th CFD Conference, June 1989.