AIAA 81-0372R

Computer-Aided Design System Geared Toward Conceptual Design in a Research Environment

Sharon H. Stack*
NASA Langley Research Center, Hampton, Va.

A computer-aided design system has recently been developed specifically for the small research group environment. The system is implemented on a Prime 400 minicomputer linked with a CDC 6600 computer. The goal was to assign the minicomputer specific tasks, such as data input and graphics, thereby reserving the large mainframe computer for time-consuming analysis codes. The basic structure of the design system consists of GEMPAK, a computer code that generates detailed configuration geometry from a minimum of input; interface programs that reformat GEMPAK geometry for input to the analysis codes, and utility programs that simplify computer access and data interpretation. The working system has had a large positive impact on the quantity and quality of research performed by the originating group. This paper will describe the system, the major factors that contributed to its particular form, and give examples of its application.

Introduction

Computer-Aided Research

THE term "computer-aided design/computer-aided manufacturing" (CAD/CAM) is used widely today in just about all technology areas from aircraft companies and architecture firms to the manufacturers of light bulbs. In most cases, the goals of these systems are the design and/or manufacture of a product or products. A distinction should be made between these types of systems and the one that will be discussed in this paper. To help clarify the difference, a more appropriate term for this paper would have been computer-aided research (CAR), defined simply as the use of computers to help the researcher in any way possible. As the acronym suggests, most researchers are inclined to think of computers as strictly a means (or necessary evil) for getting from point A to point B or, in other words, from "concept" to "proof of concept" as quickly as possible. The researcher generally has neither the time nor the interest to become proficient in the use of computers. He learns what is necessary for the sake of economy and expediency. In addition, the researcher does not have a predefined set of procedures that are sequentially performed, particularly in conceptual design where he is free to make gross modifications to his concept and pursue a totally different logic path. The CAD/CAM systems are designed to coordinate the efforts of several people or groups of people (designers, lofting engineers, aerodynamicists, propulsion engineers) working on a particular design or design problem. In a research environment, the inverse is often the case. One researcher may be working on several projects at one time and having to correlate, analyze, and modify each of them through many of the discipline areas himself. To add to this task the chore of having to become proficient at utilizing the computer(s) in all areas is an unreasonable requirement and hardly practical.

The fact remains, however, that if a researcher is able to use the computer himself instead of having to depend on others to do it for him, he will get his analysis results much faster. Therefore the goal of a CAR system is to allow a researcher, regardless of his computer expertise, to use the computer as an effective tool in his work.

The Research Group

The environment in which a computer-aided system will be used and the types of applications are important factors in the initial stages of its design. The Hypersonic Aerodynamics Branch at NASA Langley Research Center, consists of 17 engineers and 3 mathematicians. The types of research conducted by the group include 1) numerical and experimental aerodynamic and thermodynamic evaluation of vehicle concepts from subsonic through hypersonic speeds; 2) engine/airframe integration technology; 3) systems analysis for performance tradeoff trends of integrated concepts (structures, etc.); and 4) conceptual design and optimization of hypersonic vehicles (cruise aircraft, missiles, etc.). The broad scope of these tasks has forced the research group to interact constantly with other organizations to compare and interchange ideas, analytical methods, and technical expertise. These other groups are more than likely in aerospace companies or other government agencies with computational capabilities different from those available at the Langley Research Center. This interaction with other groups and the fact that our own group is composed primarily of researchers and not programmers were primary considerations in the development of our current analysis system.

The remainder of this paper will discuss the in-house development of ACTION, a computer-aided research system and will describe some aspects of the system in detail.

ACTION, A Computer-Aided Research System

Geometry Definition

Through the years the Hypersonic Aerodynamics Branch research group has become increasingly dependent upon the computer for the majority of its computational analysis. The old "lots of hand-generated input into a large mainframe computer to grind out reams of tabulated data" method had already been superseded by "a minimum of hand-generated input into a large mainframe to grind..." method. The problems encountered with the old "hand-input" method had been eased considerably by the in-house development of GEMPAK,[1] a computer code that generates detailed configuration geometry with a minimum of input.

GEMPAK

With GEMPAK, a fuselage can be analytically defined with continuous and/or discontinuous lofting curves with a minimum of input required from the researcher. Figure 1a illustrates the input that would be required to generate the curve and cross sections. The input points include the end points of each curve segment and the segment slope control point or the intersection of the slopes tangent at the end points. Planar surfaces or winglike geometry can be generated with basic input parameters such as aspect ratio, taper ratio,

Presented as Paper 81-0372 at the AIAA 19th Aerospace Sciences Meeting, St. Louis, Mo., Jan. 12-15, 1981; submitted March 4, 1981; revision received June 22, 1981.

*Aerospace Engineer, Performance Aerodynamics Branch, High-Speed Aerodynamics Division. Member AIAA.

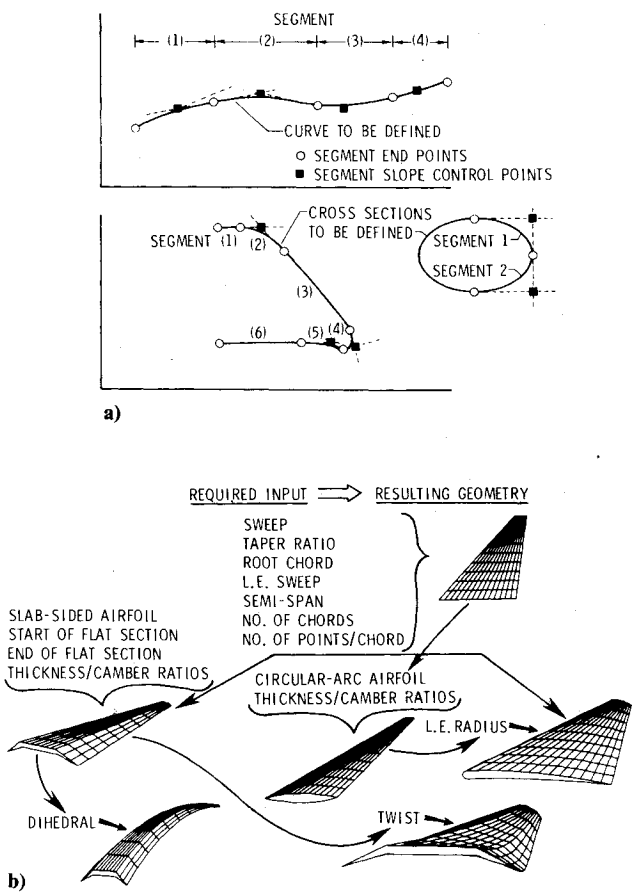

Fig. 1 a) GEMPAK lofted input. b) GEMPAK planar surface generation.

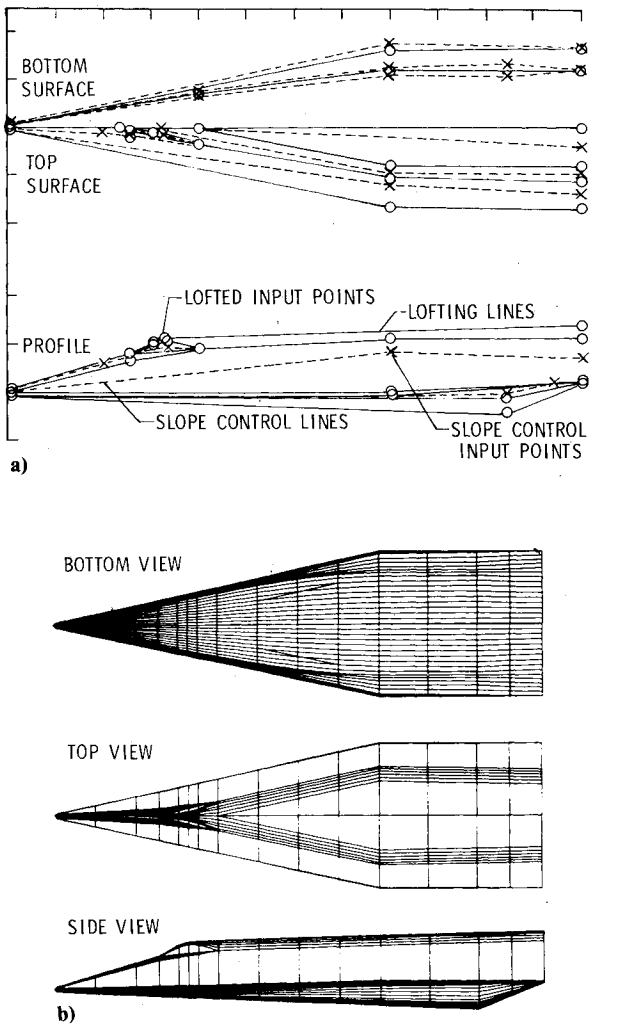

Fig. 2 a) Fuselage lofted input to GEMPAK. b) Resulting GEMPAK-generated geometry.

sweep angle, and airfoil shape. Other GEMPAK options for planar surfaces, such as change in dihedral, twist, etc. are shown in Fig. 1b. Figure 2 shows an example of lofted fuselage input and the resulting geometry generated by GEMPAK. Once the geometry input is defined the user is free to specify the amount and location of detail in the final geometry. Several views of the same configuration in Fig. 3 illustrate this enrichment of geometry by simply increasing the number of points and their location in each cross section. After all component geometries are generated, GEMPAK scales the geometries for compatibility and merges all components into the complete configuration (Fig. 4).

Geometry Interfaces

The development of interface programs that arrange the geometry generated by GEMPAK into formats specified by several aerodynamic programs eliminated the stringent requirement that the user redefine his configuration geometry for each program he wishes to use and then repeat the process for any eventual modification of his vehicle. For example, the Hypersonic Arbitrary-Body Aerodynamic Program[2] requires that the fuselage geometry be input as a series of four-sided panels, the user having to input the coordinates for each of the corner points. The Harris Wave Drag Program[3] accepts fuselage cross-section coordinates at constant body stations. The lifting-surface[4] and the vortex-lattice[5] programs both require planform and camber definitions, but each specifies a different arrangement or form for the input. Having to exercise all of these programs to predict the aerodynamic performance of a vehicle concept, the researcher ends up wasting a large portion of his time juggling and rejuggling geometry coordinates. Thus the concept of using a single geometry generation program and interfaces to supply geometry information to other programs (Fig. 5) has succeeded in removing much of the tedium from input preparation.

The initial mode of operation, however, was in the batch mode; that is, a job was submitted to the computer and the results returned to the user hours later. Often, to expedite this procedure, the researcher would submit his job to generate the detailed geometry, get a computer drawing of the resulting geometry, interface the geometry to an analysis program, and execute the analysis program all in one step. Needless to say, this "brute-force" method resulted in an unnecessary waste of computer resources and valuable time, but an occasional success at a crucial time was enough to keep the practice going. What was really needed in this procedure was the capability of quickly verifying geometry graphically before performing any analysis.

Interactive Graphics

Although interactive computer capability has been available for a number of years, the researcher is often encumbered by limited access to it and excessive costs. But even if these two problems are minimized, the resulting increase in numbers of users communicating with the computer in a time-sharing mode slows down the overall response time. If a user has to wait 10-15 min for his model geometry to be generated and a drawing of the geometry to appear on the screen of a graphics terminal, such as usually occurs within the confines of a large computer complex, then interactive computer design rapidly becomes ineffectual, not to mention

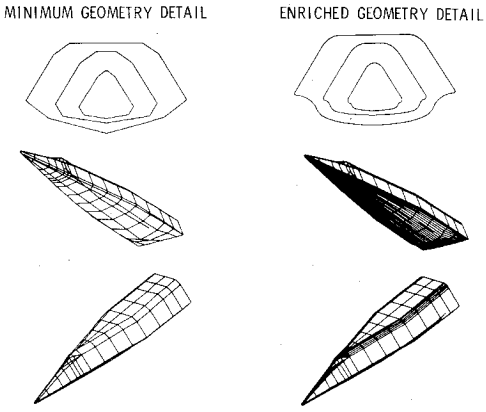

Fig. 3 User control of geometry detail.

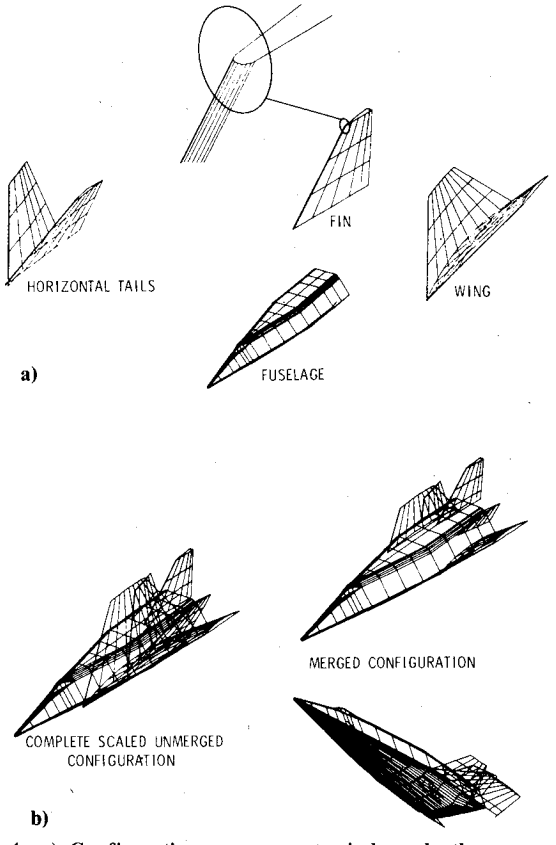

Fig. 4 a) Configuration components independently generated. b) GEMPAK scaling/merging capabilities.

frustrating, to the researcher. He will choose instead to perform his analysis in the batch mode operation previously outlined.

Distributed Computing

In an effort to relieve this and other similar situations, the computer center at Langley embarked on a project of evaluating a distributive system in a production interactive graphics atmosphere. The system consists of a Prime 400 minicomputer with a communication link to a Control Data Corporation 6600 computer and is accessed by eight remote terminals, each of which was dedicated for the use of a research group. The goal for the project was to assign the minicomputer specific tasks such as graphics and data input where speed and response time are primary considerations while accuracy and "compute power" are secondary. In this way the large mainframe computers can be reserved for the tasks for which they are best suited—"number crunching" in

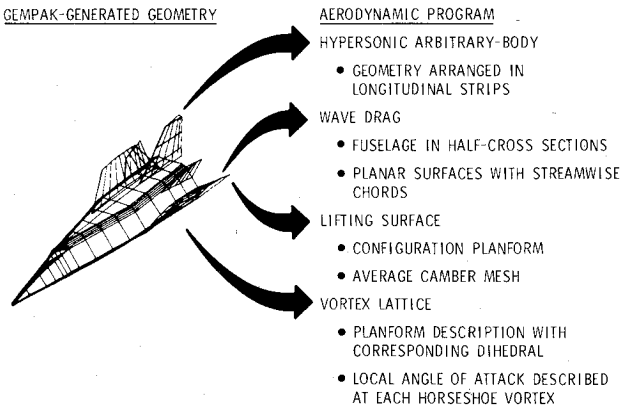

Fig. 5 Geometry interfaces to aerodynamic programs.

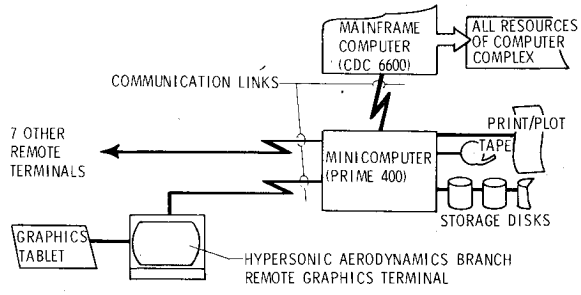

Fig. 6 Distributed computing configuration.

an uninterrupted mode—and the minicomputer could be used to satisfy the researchers need for quick-response graphics. Figure 6 shows in a simplified diagram the various components of the distributed system. Note that the direct link to the CDC 6600 mainframe computer gives any job submitted from the minicomputer access to the entire computer complex including other mainframes, printers, plotters, file storage devices, etc.

The relationship between the mainframe and the minicomputer is similar to that of a computer with a remote batch terminal. As illustrated in Fig. 7, the communication link acts as a "card reader" for jobs going from the Prime to the CDC machine. A file being transmitted in this direction will look exactly like a card deck, that is, a series of 80-column source images. For jobs being returned to the minicomputer after completion on the mainframe, the link becomes a "line printer" and transmits 136-character source images. Jobs submitted via the minicomputer are assigned a higher priority on the mainframe system than other batch jobs, since the assumption is made that the submitter is working in an interactive mode and is waiting at a terminal for his results so he can continue with his analysis process.

It was the access to this distributive system that provided the incentive for the Hypersonic Aerodynamics Branch to update its analytical procedure with interactive techniques.

Assessing Existing Software

Now that hardware was available that made interactive analysis feasible, all that remained for the Branch to be in production was to implement the necessary software on the distributive system. The choices of how to go about this narrowed down to 1) using interactive techniques already developed and in use or 2) developing new techniques. Because of the limited programmer support available to the Hypersonic Aerodynamics Branch, the former route appeared the most attractive. The subsequent survey of existing computer-aided design systems, however, revealed that many of the CAD-CAM systems used interactive software packages that were specialized in application and required trained

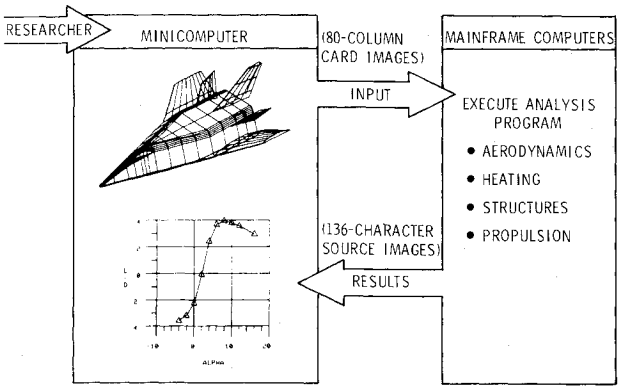

Fig. 7 Minicomputer-mainframe relationship.

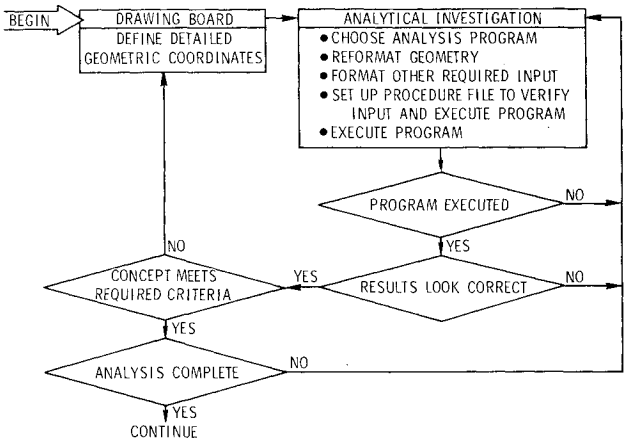

Fig. 8 Bottle necks in computer-aided research.

operators and expensive equipment such as refresh graphics terminals and dedicated computers. These systems were for the most part too big, too sophisticated, and/or too costly to operate for conceptual research and design. The smaller and easier to use systems that were available were either so inflexible as to applications or so computer system-oriented that they were impractical for the group needs because the interaction with other groups would be too severely restricted. The Hypersonic Aerodynamics Branch research goals and mode of operation seemed to fit precisely in the middle of this apparent gap between the giant CAD-CAM systems and the smaller highly tailored systems.

System Development

Finding no other recourse, the group undertook the development of an intermediate-type system that would speed up the analysis procedures and would take full advantage of the distributed computing philosophy.

Approach

The analysis procedures already in batch use were examined in order to identify the real "bottle necks" and decide what could be done about them. The bottle necks indicated in Fig. 8 that most affected the research and researchers included the handling and verification of data (input and output) and the operations required to access the computer and exercise the desired analysis program. Not only are these areas the most tedious and time-consuming, but they are also loaded with "user pitfalls"; that is, any errors are most likely to occur in the input phase, in the interpretation of the output, or in the access to the computer. It became the major goal of the entire approach to developing a computer-aided research system to overcome these problem areas. To solve these problems efforts were focused on interactive graphics techniques.

An early decision that had to be made was how to distribute the work load on the distributive system; that is, which computer should do which work and what information should be passed between the computers. It was obvious from the beginning that an attempt to modify all of the analysis codes for the minicomputer would not only be a horrendous undertaking, but would require a battery of programmers just to maintain and update the system. With this in mind, it was decided to keep the majority of the analysis codes resident on the mainframe computer. Not only would this approach take full advantage of distributed computing, but it would avoid having to maintain the analysis codes on both the CDC and Prime systems. This feature also permits the research group to take immediate advantage of new computer codes developed at Langley and elsewhere without having first to adapt the codes to the minicomputer.

Basic Structure

As previously mentioned, the branch had already overcome many of the pitfalls of data manipulation by using a single geometry generation program (GEMPAK) and interfaces to supply geometry information to other analysis codes. This concept lent itself well to distributed computing. All geometry input, generation, and modifications that require user intervention could be performed on the minicomputer and verified with interactive graphics techniques. The final geometry could then be transmitted together with other input—flight conditions, for example—to the mainframe computer where the information is converted to the format required by the desired analysis code; the code is executed; and the results are transmitted back to the minicomputer for graphical interpretation. In this way, the researcher would work directly only with the minicomputer. The entire network of these programs and utilities is called ACTION.

There are two ways of thinking of the system ACTION. The first is that of a physical package of routines that another research group doing similar work could install on their computer and use. The second and more important is to think of ACTION as a philosophy or a way to organize the analysis codes, graphics packages, and other utilities with which a group of users is already familiar and to make these computer programs more easily accessible to the researcher. ACTION consists of not a single program but many small programs that allow the user to choose his own logic path through a series of analysis programs without having to worry about computer commands and arranging his data in the proper sequence. Figure 9 exemplifies how ACTION queries the user through his chosen logic path and performs the necessary

Table 1 Typical ACTION programs/utilities

MANAGE	Creates a directory file that contains general information about a user's job such as title, configuration components and their status
GEOMIO	Read and writes configuration geometry from/to designated binary files
GEMT38	Creates the geometry source file that will be transmitted to the mainframe computers
XNOSAP	Collects the necessary information from the user to execute an analysis program on the mainframe computers
GCCNOS	Generates the mainframe control cards necessary to execute the analysis program
S2COM	Generates the minicomputer control cards to collect the necessary information together and submit the job to the mainframe
COMSTA	Allows user to interrogate the system as to whether his job has reached completion on the mainframe and results been returned to the minicomputer
XDSPLY	Permits user the option of graphically displaying his results, printing the results or both

(>) INDICATES USER INPUT

IN THIS EXAMPLE, THE USER HAS ALREADY GENERATED CONFIGURATION GEOMETRY AND GRAPHICALLY VERIFIED IT. HE NOW WISHES TO EXECUTE AN ANALYSIS PROGRAM WITH THIS GEOMETRY AS INPUT.

ACTION PERFORMS THE FOLLOWING TASKS DURING THIS SESSION

- PREPARES THE GEOMETRY AS INPUT TO THE CORRECT INTERFACE
- GENERATES THE CONTROL CARDS TO EXECUTE THE CHOSEN ANALYSIS PROGRAM AND DISPOSE THE RESULTS ACCORDING TO USER OPTION
- PREPARES ALL INPUT TO BE SUBMITTED (CONTROL CARDS, GEOMETRY, AND ADDITIONAL INFORMATION)
- SUBMITS THE JOB TO THE MAINFRAME

Fig. 9 Example ACTION user session.

operations and procedures, all of which are invisible to the user. To more fully describe the types of operations that ACTION does for the user, Table 1 lists some ACTION routines and describes their functions.

The general flow of ACTION on the minicomputer is illustrated in Fig. 10. The boxes solidly outlined indicate the basic structure of ACTION: the geometry generation program GEMPAK and utilities to read and store geometry and analysis results. The dashed boxes indicate the areas where new independent capabilities can be included.

With a system such as this, there is literally no limit to the number of programs or their form that ACTION can handle. For example, the program GEMPAK has been modified to execute on the minicomputer, but was initially written to operate in a batch mode—its input in formatted card images. Working in an interactive mode, however, typing in card images at the keyboard is ridiculous and was soon replaced with a program that allows the user to digitize his lofted drawing or sketch from a graphics tablet. This digitized information is then automatically formatted as input to GEMPAK, so that detailed configuration geometry can be generated. This digitizing program was written by a researcher totally unfamiliar with ACTION. To incorporate it into the ACTION system was simply a matter of including the program title as a user-option in an existing ACTION menu and including the code necessary to locate the program and to execute it. At the completion of the digitizing program, instead of stopping execution, control is returned to ACTION so that the user can choose his next step. This digitizing program would become an option in the box labeled "Input Geometry by any Method" in Fig. 10. The flow from that program would continue to "Convert to GEMPAK Input" and then to the GEMPAK program. There is no restriction on how geometry is input or generated as long as it is stored properly. The same is true of subsequent ACTION capabilities. The modularity of the whole system makes ACTION easy to modify and update.

Human Factors

In order for ACTION to be a useful tool, the researcher and

his attitudes and reactions had to be considered throughout the development. It is possible to develop a system that does all sorts of wonderful things, but if it is not tailored to the working environment in which it will be used and to the researchers who will be using it, it is worthless. The researchers should be consulted even before development begins. New procedures should be tested by a researcher for his comments or suggestions and the system must be flexible enough to incorporate his ideas quickly. Some user suggestions or complaints may have nothing to do with the actual function of a code. For example, sitting in front of a blank terminal screen for 15 s can seem like an eternity. Just a message that something is being done printed on the screen can reassure the user that, no, the computer has not broken down and that, yes, the computer has received his last command and is working on it. One of the ACTION programs that is used to curvefit and display aerodynamic data initially plotted the calculated curve and then the actual data points. The order in which these were plotted brought the immediate comments from the users that it looked as though the data were being "fudged." The programmer reversed the order in which the information was being plotted to the more natural sequence: data points first and then the faired curve, and the complaints ceased. To a programmer this detail may seem unimportant and irrelevant, but to a user who has to work with it daily, it can be distracting to the point of becoming a nuisance.

The saying "you can't please everyone all the time..." is especially true in the design of a computer-aided system, but the effort should be made to try by giving users choices of how to use the system rather than structuring a rigid procedure. One of the geometry options in ACTION allows the user to input a point-by-point description of a fuselage by digitizing cross sections. Rather than stipulating how this must be done, the user is only instructed to digitize the points. When he has finished, he is asked whether the points were digitized from top to bottom or from bottom to top. By doing this the user is free to work in the way most natural to him.

Another example of allowing the user freedom to customize an application to his own specific needs is illustrated by the flexibility of the plotting program previously mentioned. The researcher can simply use this program to plot his data and allow the program to determine scales, symbols, etc. or he can specify the scales, symbols, line types, and tic mark locations himself. He can have several sets of data plotted on the same graph for comparison and can zoom in on areas of the graph for better detail depending on his requirements. Figure 11 shows some of the user options available. Most of these capabilities were added to the program at the requests of users.

Another important feature of such a system is that no one error that a user might inadvertently make, such as typing in a wrong input, will be catastrophic. That is, the user must be made to feel confident that, in spite of an error he makes, he will always be able to recover the work previously done. It is the responsibility of the programmer to foresee possible user errors and provide the code to either prevent them from happening or to safeguard against their consequences.

A user being made to feel at ease working with interactive computer aids regardless of his computer expertise can mean the difference between a system that is seldom used and one that becomes a useful tool.

Applications

When ACTION was initially outlined, the potential of interactive graphics was not fully realized. The prospect of being able to interactively input configuration geometry and verify it quickly with graphics was enough to make the development effort worthwhile. However, once the system was operational, the users themselves found numerous applications that helped them in their work. For example, a missile concept, shown in Fig. 12 had a design constraint that

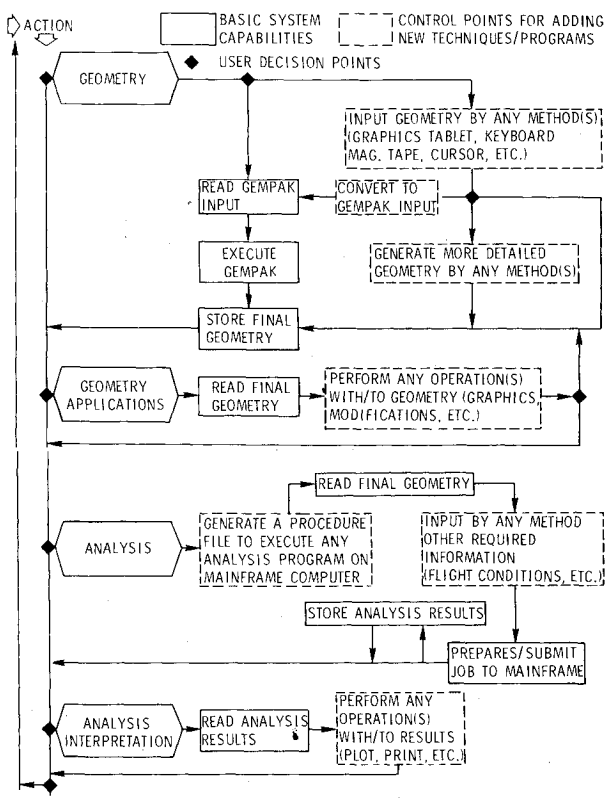

Fig. 10 ACTION modularity.

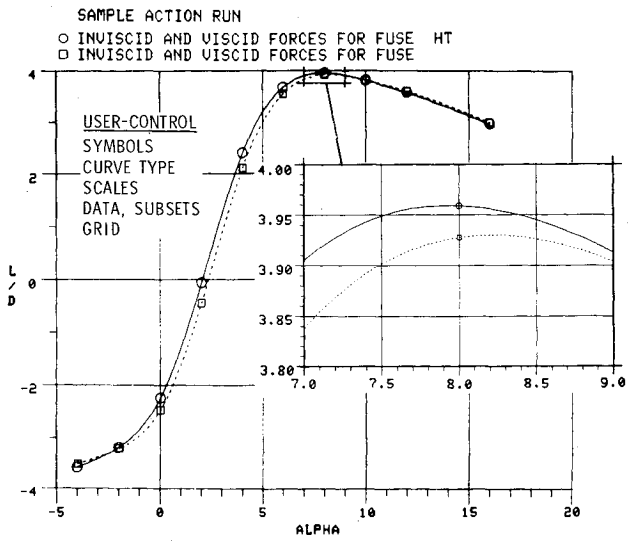

Fig. 11 User control of graphics.

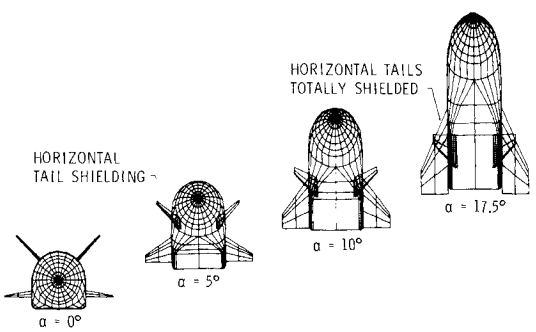

Fig. 12 Application of visual optimization.

restricted its size and total span. The researcher was able to use interactive graphics to visually optimize his vehicle geometry by taking into account the component shielding that occurred during a mission. As the missile increases its flight attitude or angle of attack, the horizontal tails are shielded more and more by the body and then the wing. Since the span of the tails is restricted, the options open to the researcher are to change the wing planform and/or to shift the tails to a less shielded position. These decisions involving the aerodynamic performance of the missile were made without executing any analysis code. This application may appear to be obvious, but it is typical of a researcher's use of graphics. The point to be made is that the system must be flexible enough to allow the user freedom to discover his own applications.

Impact

The ACTION system had only been operational for a few months before it became the standard mode of geometry generation and verification within the research group. Even researchers who in the past have depended on others to perform the access and execution of computer codes for them are now enthusiastic ACTION users. The system has not only decreased the time required for analysis, but has actually reduced the total computer use of the group, because the researcher can now spot errors or inconsistencies in his data at intermediate control points and can abort or modify the job before completion, thus avoiding unnecessary computer runs.

In addition to being able to analyze more concepts in a shorter amount of time, configurations and ideas that were impossible to consider before can now be pursued without having to worry about the amount of time and effort that will be required.

Future Work

The personnel of the Hypersonic Aerodynamics Branch feel that the surface has only been scratched with the graphics techniques currently in use in ACTION. Methods are being developed for visually scanning large amounts of data such as the techniques described in Ref. 6 where aerodynamic, thermodynamic, and structural analysis data are graphically overlayed on the vehicle geometry to highlight, for example, regions of high drag, high temperatures, or high stress. Color graphics used in much the same way has already been investigated (Ref. 7) and proven successful enough to justify the purchase of color hardware. It is planned to use color graphics to represent three-dimensional flowfield data.

The list of tasks for improving the ACTION system continues to grow as the research group becomes aware of more and more application areas, from model fabrication by numerical control machines to using interactive graphics for report figures.

Developing new analytical tools is as much a part of research technology as designing a vehicle concept to fly Mach 5 and can contribute as much. But care must be taken as in any technology area to avoid duplicating efforts or "reinventing the wheel." The researcher should attempt to keep abreast of computer hardware and software developments and new applications, and to share his own ideas and achievements.

Conclusions

The ACTION computer-aided research system has been readily accepted by all of its users. It is easy to use (does not require computer expertise to operate) and it is relatively hardware independent. The task of defining a numerical model of a vehicle concept and predicting its performance can be accomplished in hours instead of days or weeks because ACTION aids the researcher in performing all the tedious operations that tend to obscure the research problem at hand. All research decisions are left to the researcher; in other words, the system does not optimize, minimize, or perform other "mysterious" operations. Particular emphasis has been placed on maintaining ease of adding new capabilities and modifying old ones, so that any suggestions or recommendations from the users can be incorporated quickly.

Demonstration of the versatility and the user-oriented characteristics of the system has caused consistently favorable responses from other independent research groups, as well as research subgroups within large companies. This surprisingly large amount of outside interest has substantiated the hypothesis that the user doing basic research is often forgotten during the development of computer programs and computer-aided systems. The experiences of the Hypersonic Aerodynamics Branch have shown that this oversight can be remedied with systems such as ACTION and that the effort involved in the development will be repaid manyfold.

References

[1] Stack, S.H., Edwards, C.L.W., and Small, W.J., "GEMPAK: An Arbitrary Aircraft Geometry Generator," NASA TP-1022, Dec. 1977.

[2] "Hypersonic Arbitrary-Body Aerodynamic Computer Program (Mark III Version)," Rept. DAC 61552, Vols. I and II (Air Force Contracts F33615 67C 1008 and F33615 67C 1602), McDonnell Douglas Corp., April 1968; Gentry, A.E., "Vol. I—User's Manual" (available from DTIC as AD851811); Gentry, A.E. and Smyth, D.N., "Vol. II—Program Formulation and Listings" (available from DTIC as AD861812).

[3] Harris, R.V. Jr., "An Analysis and Correlation of Aircraft Wave Drag," NASA TM X-947, March 1964.

[4] Carlson, H.W. and Middleton, W.D., "A Numerical Method for the Design of Camber Surfaces of Supersonic Wings with Arbitrary Planforms," NASA D-2341, June 1964.

[5] Margason, R.J. and Lamar, J.E., "Vortex-Lattice FORTRAN Program for Estimating Subsonic Aerodynamic Characteristics of Complex Planforms," NASA TN D-6142, Feb. 1971.

[6] Talcott, N.A. Jr., "A Computer Graphics Display Technique for the Examination of Aircraft Design Data," AIAA Paper 81-0370, Jan. 1981.

[7] Edwards, C.L.W., Meissner, F.T., and Hall, J.B., "The Use of Computer-Generated Color Graphics Images for Transient Thermal Analysis," NASA TP-1455, July 1979.