LINDA HALL LIBRARY
5109 CHERRY STREET
KANSAS CITY, MISSOURI
64110-2498
PHONE (816) 363-4600
FAX: (816) 926-8785

6/12/03 DocServ #: 161072

9:00

SHIP TO:

Attn: Garvin Forrester
Calmar Research
63 E. Genesee St.
Baldwinsville NY 13027
USA

Fax:
Phone: 315-626-6800
Ariel:
Email: g.forrester@calmarresearch.com

Shelved as:

Location:

Title: AIAA

Volume:

Issue: AIAA-87-2902

Date: 1987

Author: D. Snepp and Roger Pomeroy

Article Title: A Geometry System for Aerodynamic Design

Pages:

Accept Non English? No

Regular

ElecDel

AIAA

This request complies with 'Fair Use'

Max Cost:

Reference Number:

Account Number:
FEDEX Account Number:

Notes:

DOCSERV / WEB / PULL SLIP

AIAA'87

AIAA-87-2902

A Geometry System for Aerodynamic Design

David K. Snepp, Roger C. Pomeroy

Boeing Commercial Airplane Company
P.O. Box 3707, M/S 6A-36
Seattle, WA 98124

AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting
September 14-16, 1987/St. Louis, Missouri

AIAA-87-2902

A GEOMETRY SYSTEM FOR AERODYNAMIC DESIGN

David K. Snepp*, Roger C. Pomeroy*
Boeing Commercial Airplane Company
Seattle, Washington 98124

A88-14265

Abstract

This paper describes an interactive, three-dimensional geometry system known as the Aero Grid and Paneling System (AGPS). Developed by The Boeing Company, AGPS is a surface geometry system focusing on the needs of engineering technical staffs such as aerodynamics. It provides an efficient means of defining any three-dimensional surface, such as that representing the external shape of an airplane. AGPS has built-in and user-programmable features for extracting geometric data in the proper form for use by computational fluid dynamics (CFD) or other analysis codes. The results of these codes can then be brought back into AGPS and displayed graphically with the surface geometry data in an integrated manner.

AGPS operates on a minicomputer or engineering workstation. It uses cubic, quintic, and rational b-spline mathematics to represent curves and surfaces. The program may be run via menus or a command language. The command language allows a complex sequence of steps to be saved and repeated quickly.

An overview of AGPS and its usage is presented. Topics include basic features of the program, the role of AGPS in the aerodynamic design process, program operation, mathematical modeling concepts, data structure, and lessons learned in the development and support of the program.

Introduction

As the aerospace industry shifts more and more to the use of computers in the definition of its products, much has been written about the subject of computer-aided design (CAD) (refs. 1 through 5). These writings, as well as the CAD systems themselves, typically focus upon the drafting or conceptual design environment. However, such systems often do not satisfy the needs of engineering specialists in technical support staffs such as aerodynamics. Effective use of computers in aerodynamic design requires a geometry system that addresses the surface definition, extraction, and display requirements of the design/analysis process in an integrated fashion.

Surface definition is the primary function of any aerodynamic geometry system. The aerodynamic designer must be able to efficiently define a complex, three-dimensional mathematical model of the outside contours of an aircraft. This involves defining surface contours, evaluating these surfaces (e.g., for smoothness), and modifying the contours as required. Geometry systems must be adaptable to unforeseen, one-of-a-kind applications. In addition, they must address more routine applications such as wing design. They must accept the typical inputs of such applications, in this case nondimensional airfoils, wing planform, and twist and shear distributions. Because of the iterative nature of aerodynamic design, the system must provide a means of storing and repeating a complex sequence of steps.

Extraction capabilities are required to fabricate or analyze surface contours. Surface data must be sent to numerical control (NC) machinery for fabrication of wind tunnel models. Also essential is a link with the aerodynamic analysis codes that have become a fundamental part of the modern design process. As described in Reference 6, "... generating the geometric input data, i.e., geometric models, for the computational codes can be a time-consuming operation, especially when models must be generated for several codes during the design exercise." Some means of storing sequences of steps required to extract and properly format geometry data must be provided. The extraction methods must be flexible enough to adapt to the individual requirements of the various aerodynamic codes, ranging from simple two-dimensional linear methods to complex three-dimensional Navier-Stokes codes.

As analysis codes become more capable, efficient display of the data generated by such codes becomes essential. The value of such displays is described clearly in References 7 through 9. Modern panel methods or computational fluid dynamics (CFD) codes generate aerodynamic data at thousands or tens-of-thousands of locations. Interpretation of such data is somewhat akin to drinking from a firehose — the shear volume of the product is overwhelming. Thus, the aerodynamic engineer requires some means of displaying computed analysis data and integrating these data with the physical and computational models used.

The Aero Grid and Paneling System (AGPS)

Program Summary

The Aero Grid and Paneling System (AGPS) was developed by the Aerodynamics Technology staff of the Boeing Commercial Airplane Company (BCAC) with support from Boeing Computer Services (BCS). It is a surface geometry system focusing on the needs of engineering technical staffs such as aerodynamics, as summarized in Figure 1. AGPS provides an efficient means of developing any three-dimensional surface or the entire external shape of an airplane. It also has built-in and user-programmable features for extracting data in the proper form for use by CFD or other analysis codes. The results of these codes can then be brought back into

*Specialist Engineer, Member AIAA

A surface geometry system for
- **Definition of external surfaces**
- **Extraction of surface data**
  - **For input to analysis programs**
  - **For N.C. fabrication**
- **Display of surface properties**

Figure 1. AGPS Overview

AGPS and displayed graphically with the surface geometry data in an integrated manner.

AGPS runs interactively on minicomputers or engineering workstations. It is currently being used on VAX and MicroVAX computers under the Digital Equipment Corporation VMS operating system. The VAXstation II/GPX workstation shown in Figure 2 allows a complete integration of the wireframe, shaded graphics, and color display capabilities of the program. As described in Reference 10, workstations offer many advantages, including lowered cost and better distribution of computing resources. In addition, the "screen window" environment available on many workstations allows multiple jobs to be run simultaneously from the same terminal.

Figure 2. AGPS Operation on Workstation

AGPS History

Initial development of AGPS took place over a relatively short period of time. Work began in January 1980, and a preliminary version of the program was available in September. This was made possible by two major factors. First, the development team consisted of a handful of exceptionally talented, motivated individuals with backgrounds in mathematics, computing, and engineering. Secondly, the team was able to draw on experience with experimental "prototype" programs developed earlier. Though not all were released to the general user-community, these programs provided a useful foundation for AGPS development.

Since its initial development in 1980, AGPS has been used throughout The Boeing Company. Usage increased dramatically in 1983 when a user-support group was formed and a VAX-11/780 was dedicated to AGPS development and support. Currently, aero staff design and analysis of the 7J7 is being done using AGPS in conjunction with the CATIA CAD system. An interface between AGPS and CATIA was recently developed using the Initial Graphics Exchange Specification (IGES) format.

AGPS — An Aerodynamic Design Tool

The Role of AGPS

AGPS functions as a tool in the aerodynamic design process. It focuses upon the three basic needs of the aerodynamic designer:

1. Accurate definition and control of the external aircraft shape.
2. Versatile data-extraction capability for analysis and fabrication.
3. Effective display of geometry and aerodynamic flow properties.

An excellent illustration of the usefulness of AGPS in meeting these three needs is the design of the wingtip pods on the E-6A. This configuration includes a wingtip antenna pod, with another pod slung from a strut beneath. As shown in Figures 3 through 5, AGPS was used to model the surfaces of each aircraft component, to prepare inputs to panel analysis codes based on these surfaces, and to display the results of the analysis codes. The modeling and paneling procedures were automated in command file form to allow numerous design variations to be analyzed in a timely manner.

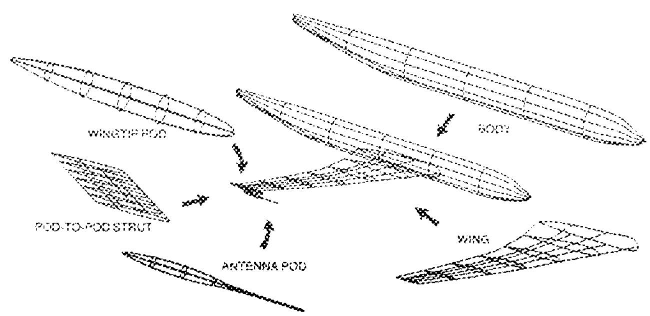

Figure 3. E-6A Surface Definition

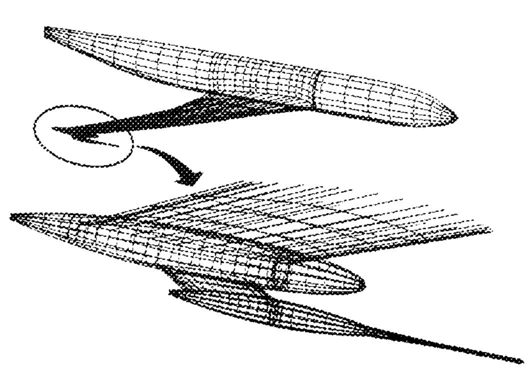

Figure 4. E-6A Panel Networks

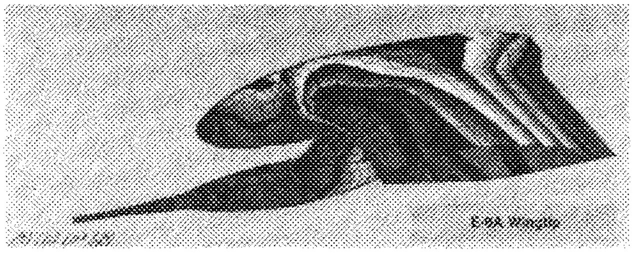

Figure 5. E-6A Display of Flow Properties

Definition of Geometry

AGPS can define virtually any type of curve or surface geometry. The various parts of a complex shape can be developed in pieces or as assembled units. Geometric data can be input from external files or generated within the program, then interactively manipulated if desired. These data can be used to accurately define and control the aircraft's external shape. The user can input or create data, act upon it with many geometry-related graphics tools, and immediately see the results in wire-frame or realistic shaded graphics form.

The program has been applied in the design and analysis of both commercial transports and military configurations. As shown in Figure 6, it has been used to model complete aircraft and a variety of aircraft components, including:

- Wings, including winglets
- Fuselages
- Engine nacelles
- Tail surfaces
- Flap track fairings
- Wingtip pods
- Radome antennas
- Cabin interior sidewall panels
- Ultrabypass engine fan blades

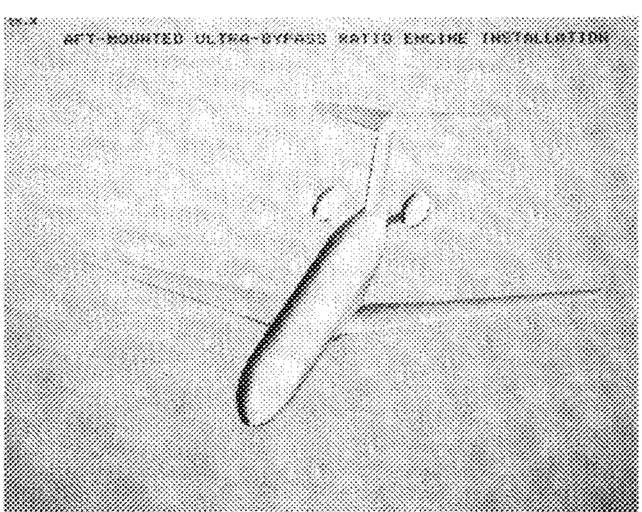

Figure 6. Geometric Modeling of Complete Aircraft

Extraction of Data

AGPS provides numerous utilities for extraction of curve and surface data. Networks of points can be easily extracted for input to aerodynamic analysis codes. Such data might define panel networks on the aircraft's external surface or a complex three-dimensional grid of points extending off of the surface. This ability to input, manipulate, and, if desired, output large quantities of data is essential to support modern CFD codes. Surface data can also be generated for fabrication of parts using NC machinery.

Basic extraction capabilities include the following:

- Planar cuts of surfaces
- Surface-surface intersection
- Curve-surface intersection
- Curve-plane intersection
- NC data extraction of surface points and normals
- Geodesic tracing for composite filament path

Display of Data

A key feature of AGPS is its extensive display capabilities. As point strings, curves, surfaces, etc., are defined, the user may draw any or all of these objects. As shown in Figure 7, they may be viewed in wire-frame form at any desired orientation. Color raster-graphics devices may be used to display surfaces in realistic shaded graphics form. Such displays can be extremely useful in checking surface quality.

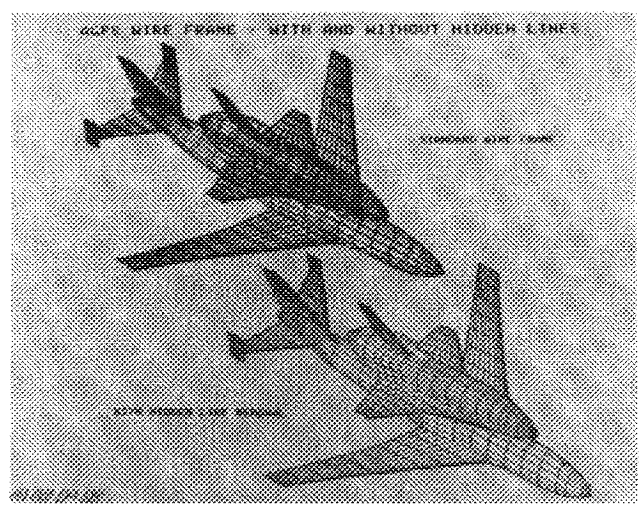

Figure 7. Wire-Frame Display

AGPS produces color-coded displays of data generated outside of the program. For example, airflow pressures calculated by an aerodynamic analysis code may be input to AGPS and displayed on a raster terminal. As shown earlier in Figure 5, the configuration appears as a pattern of colors, each color representing a range of pressure values. Display techniques for such images are very similar to those used for wire-frame and shaded graphics, allowing for an integrated display of geometry and analysis data. Note that in Figure 5, surface panel networks have been superimposed upon the configuration to better display the computational model. Figures 8 and 9 show the display of off-body pressures and streamlines along with shaded graphics images of the configurations analyzed.

Graphics terminals may be used to display the following:

- Standard wire-frame images of points, curves, or surfaces
- Wire-frame images with hidden-lines removed
- Shaded graphics images of surfaces
- Curvature plots of curves
- Three-dimensional, color-coded images of surface curvature
- Color-coded images of surface properties (e.g., pressures)
- Flowfield properties (e.g., streamlines)

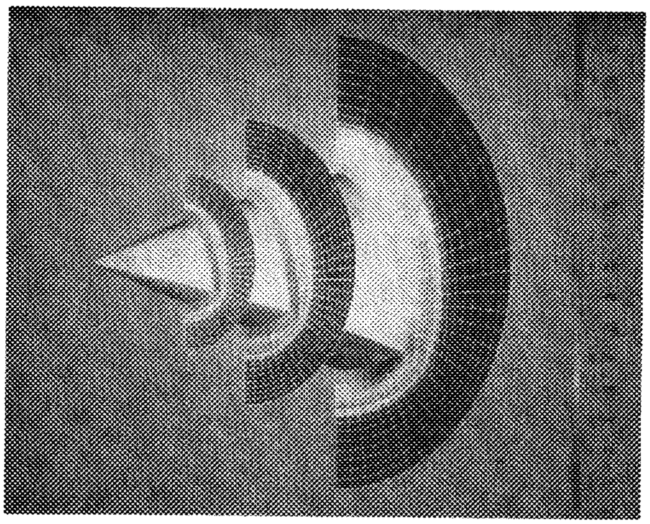

Figure 8. Display of Surfaces and Off-Body Pressures

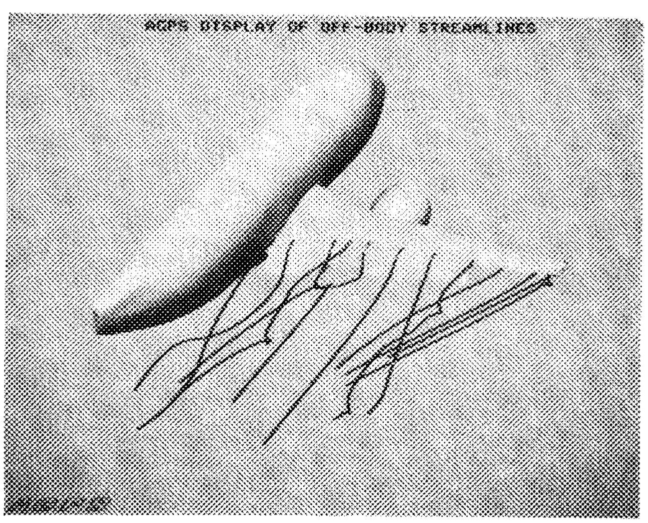

Figure 9. Display of Surfaces and Off-Body Streamlines

Program Operation

User-Interface Concepts

The AGPS user interface was designed with an emphasis on flexibility, allowing both interactive and batch execution from either graphics or nongraphics terminals. Interactive graphics operation supports both menu and type-in modes, as shown in Figure 10. The command type-in mode allows interactive execution via a keyword-oriented language including FORTRAN-like constructs such as DO-loops and IF-THEN-ELSE statements.

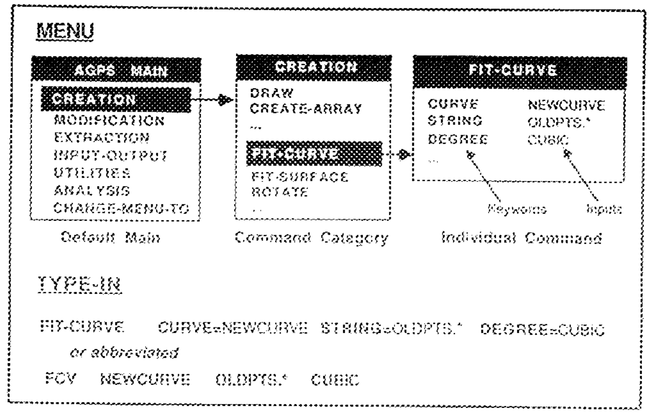

Figure 10. AGPS Menu and Type-In Modes

A command file mode makes use of the command language to combine a number of individual commands into a single routine. Command files are easy to use, since their syntax is identical to that used in the type-in mode. A complex sequence of steps can be saved and repeated easily with new input data. A journal-file capability within AGPS can be used to automatically record a user's interactive steps in command file form. Alternatively, command files can be created outside of AGPS using any text-editing utility. Command files allow new users to perform complex operations with a minimum of training. Users need only understand the required inputs and outputs of the command file, not the details of the program operations involved.

Tailoring the Program

The flexibility of the AGPS user interface allows the program to be tailored to individual applications without changing the basic program itself.

This tailoring is done by "expert users" who are proficient in the application of AGPS, not by programming analysts. These users develop specialized command files and organize them by means of specialized menu files. From the basic default menu, the average user can then call up these specialized menus to reconfigure the appearance of AGPS to suit his particular task.

A number of task-oriented AGPS "packages," consisting of a set of command files, menu files, and appropriate documentation have been prepared. These address topics such as wing lofting, panel network editing, and aerodynamic flow-property display. User response to these packages has been very positive, since they allow the average user to complete his task with a minimum of knowledge of AGPS itself.

Overview of AGPS Mathematics

Basic Concepts

AGPS uses as mathematical constructs the point, curve, and surface. (Some preliminary work has been done on solids as well.) Curves are modeled as a continuous chain of segments; surfaces are modeled as a network of four-sided patches. In most cases, the user may think of curves and surfaces as single entities. Segment and patch elements are available for manipulation by advanced users.

Geometry may be defined using either polynomial or nonuniform rational b-spline mathematics. Curves and surfaces are fit to input data points using linear interpolation, cubic or quintic splines, or quadratic- or cubic-b-splines. Continuity of desired degree can be specified across curve segment or surface patch boundaries.

Parametric Representation

Curves and surfaces are represented using parametric techniques, as illustrated in Figure 11. In the case of a curve, the X,Y,Z coordinates are converted and stored as a function of a parametric variable, S. This parametric variable is analogous to nondimensional arc length; it has a value of 0 at one end of the curve and 1.0 at the other end. Surfaces are represented as a function of two parametric variables, S and T; these vary from 0 to 1.0 in orthogonal directions along the surface.

A portion of a curve or surface may be represented by a "subrange" object. As illustrated in Figure 11, for example, a subrange curve may lie on a surface, or a subrange surface may lie on another surface. In the limited region of the subrange, these objects are mathematically equivalent to their parent. They have their own parameterization scheme, which is mapped into that of the parent object.

Parametric techniques offer several advantages. First, they remove most restrictions on curve or surface shape. For example, double-valued curves are possible, as are curves that loop or double back upon themselves. Secondly, they permit very rapid evaluation of surface data. Given S and T values, surface position and outward normal may be evaluated quickly. This is important when extracting large quantities of data, such as those used in defining surface panel networks for aerodynamic analysis. Other extraction methods within AGPS allow curves or surfaces to be evaluated at particular values of spatial coordinates X,Y, or Z.

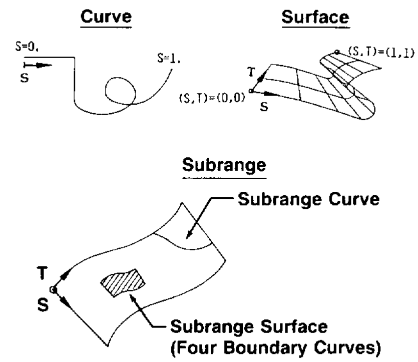

Figure 11. AGPS Parametric Representation

AGPS Data Structure

Overview

The function of the data structure is to store all of the objects read into AGPS or created by the user during a particular AGPS session. It gives the user access to the mathematical building-blocks defining each object. At any time during an AGPS session, the data structure of the program can be saved. This allows the user to exit the program, reenter at a later time, restore the data structure, and resume the session where he left off.

The AGPS data structure currently supports over 30 object types. These include geometric entities, such as point-strings, curves, and surfaces, as well as nongeometric entities, such as text objects. Nongeometric "list" objects are used to conveniently group together similar objects. For example, all surfaces representing a complete airplane configuration might be stored in a list.

The AGPS data structure has been designed to easily accept new object types. The addition of a new object or mathematical form requires only two new routines: one to create the object, the other to evaluate it. This has allowed the list of available object types to grow as dictated by new program applications.

Object Storage

AGPS uses the concept of virtual memory for an in-core data structure. Each object in the data structure is assigned an arbitrary name, either by the user or automatically by the program itself. As illustrated in Figure 12, each object is stored using one of two methods, depending upon the particular object type. These are termed "call-by-value" and "call-by-reference" objects.

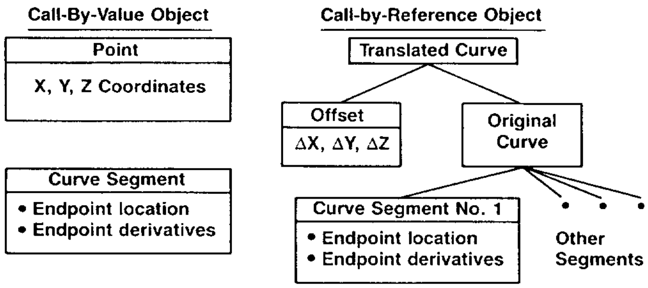

Figure 12. AGPS Data Structure

Call-by-value objects are stored independently of other items in the data structure. All of the information needed for its evaluation is stored in the object's own data block. The advantage of this type of object is that it can be evaluated very rapidly.

Call-by-reference objects are stored in terms of pointers to other objects in the data structure, termed "ancestors." While slower to evaluate, these objects offer two advantages over call-by-value objects. First, they provide for automatic regeneration. That is, when a component or ancestor of an object is modified, the object itself is automatically modified. Secondly, they allow for a very compact data structure. A call-by-reference object requires very little space in the data structure since it is stored only as a set of pointers to its ancestor objects.

Lessons Learned

The development and support of AGPS has yielded a number of useful lessons regarding geometry systems and their use by engineering technical staffs. These may be of value to anyone planning to undertake the development of such a system.

Program Features

1. Program operation via a command language allowing command files is essential. Since the command files extend the program's capability without altering the program itself, they allow the entire technical community to participate in the development of AGPS.

2. The ability to store and reexecute a sequence of steps is invaluable. Being able to alter a configuration and quickly repeat the process of panel-network extraction has allowed AGPS to support rapid turnaround in the design/analysis cycle of unusual and complex configurations.

3. The ability to record a sequence of steps during an interactive session is very important. In many cases, this feature has been used to develop AGPS procedures in real-time, "on the fly." In addition, it has proved useful in assisting error-prone users to determine where they went wrong.

4. Specialized, task-oriented menus are extremely useful in tailoring a program to the user's needs. The AGPS "packages," including commands and command files accessible via special menus, have been enthusiastically received by users.

5. A flexible, expandable data structure is extremely valuable. It allowed AGPS to be made available in a short period of time using only the basic object types, while at the same time providing an easy means of growth as the need for additional types arose.

6. An effective method of error handling is crucial to user satisfaction. This may be a luxury to be overlooked in the initial stages of program development, but provisions must be made to eventually incorporate it into the code.

Program Development

1. Effective software can be developed in a short amount of time by the right people in the right environment. The original version of AGPS was available in just 9 months, in part because the developers were creative, innovative individuals with an intimate knowledge of programming concepts and hardware.

2. In developing software, the needs and background of the user must always be kept in mind. In most cases, the developers are not the users. User feedback early in the developmental process is important; rapid response to this feedback helps to encourage acceptance of the program.

3. Tradeoffs of performance versus portability must be weighed carefully. Hardware is subject to frequent change and improvement, and a machine-specific code is difficult to support.

4. In a language-based system, careful thought should be given at the outset to basic syntax conventions. Any inconsistencies will be a constant source of aggravation and will inhibit user acceptance of the program.

Program Support

1. Software support must involve a team approach utilizing the coding skills of the programming analyst as well as the applications skills of the engineering expert-user.

2. Program support requires a different mix of skills than program development. In the support role, patience and the ability to communicate with users are more important than the innovative, entrepreneurial skills of the software-development "whiz."

3. Proper documentation must be made available to program users. The documentation must address the application of the program to tasks encountered by the users. It must be easy to use and it must be kept up to date.

4. A training program must be available to users. Like documentation, training must address the application of the program. It must be available on demand.

Conclusion

Since its introduction in 1980, AGPS has proved to be a useful tool in the aerodynamic design process. From the outset, the program was designed with an emphasis on flexibility — flexibility of application and flexibility of operation. It has allowed engineers to model a variety of aircraft configurations, to extract the geometry data needed by modern aerodynamic analysis codes, and to effectively display the data produced by these codes. By permitting both menu and command language operation, the user interface has allowed AGPS to be tailored to specific tasks without loss of versatility. During the past seven years, AGPS has demonstrated its ability to adapt to the rapidly changing needs and computing environment of the aerodynamic engineer.

Acknowledgments

The engineering supervisor responsible for the initial development of AGPS was Arvel E. Gentry. The lead engineer was Alan K. Jones, who also provided the mathematical methods used in the program. The lead programmers and primary architects of the first version of AGPS were Lloyd Tracy and Ed Hamrick.

Data and photos shown in this paper were provided by various groups in the BCAC Aerodynamics Technology and Boeing Aerospace Company Flight Technology organizations. Special thanks are offered to Brian Roskuski and Kevin Smit of the Geometry Methods group.

References

1. Feder, Aaron, "Computer Graphics Create the New Wave of Design," Astronautics and Aeronautics, June 1979.

2. Zadarnowski, John H., "CADD on the F-18 Program," Astronautics and Aeronautics, March 1980.

3. Harris, David H. W., "Applying Computer Aided Design (CAD) to the 767," Astronautics and Aeronautics, September 1980.

4. Raymer, Daniel P., "CDS Grows New Muscles," Astronautics and Aeronautics, September 1980.

5. Littlefield, J. E., and Howard, R. L., "Computer Graphics Win Over Engineers," Aerospace America, January 1987.

6. LaBozzetta, Walter F., and Cole, Paul E., "Interactive Graphics for Geometry Generation — A Program with a Contemporary Design," Journal of Aircraft, Vol. 22, No. 12, December 1985.

7. Carmichael, Ralph, and Gregory, Thomas, "Visual Aerodynamics," Astronautics and Aeronautics, June 1982.

8. Carmichael, Ralph, and Gregory, Thomas, "Interactive Computer Graphics Why's Wherefore's and Examples," Astronautics and Aeronautics, April 1983.

9. Carmichael, Ralph, and Gregory, Thomas, "Interactive Computer Graphics Getting the Picture Through Computer Graphics," Astronautics and Aeronautics, April 1983.

10. Kidwell, George H., "Workstations Take Over Conceptual Design," Aerospace America, January 1987.