

330-02
9-14

**Conversion of the Aerodynamic Preliminary Analysis System (APAS) to an
IBM PC-Compatible Format**

**John M. Kruep
Walter C. Englund**

**Space and Atmospheric Sciences Program Group
Space Systems and Concepts Division
Vehicle Analysis Branch**

Abstract

The conversion of the Aerodynamic Preliminary Analysis System (APAS) software from a Silicon Graphics UNIX-based platform to a DOS-based IBM PC-compatible is discussed. Relevant background information is given, followed by a discussion of the steps taken to accomplish the conversion and a discussion of the type of problems encountered during the conversion. A brief comparison of aerodynamic data obtained using APAS with data from another source is also made.

Introduction

The Aerodynamic Preliminary Analysis System (APAS) is a software package that can be used to perform a wide range of aerodynamic analyses. APAS is a family of FORTRAN programs, and is used by the Vehicle Analysis Branch of NASA Langley, among others. It allows the user to input a wide range of aerospace vehicle geometries, and can then analyze those geometries in the subsonic, supersonic, and hypersonic flight regimes. Given a set of test conditions (such as Mach number, altitude, center of gravity location, surface roughness, rotation rates, and angle of attack or sideslip sweeps), the APAS system can provide the user with all six aerodynamic force and moment coefficients, as well as estimates for a number of their derivatives. In addition, APAS also allows the user to define control surfaces and estimate their effectiveness.

The APAS system consists of three separate programs. The first program is APAS itself, which is used to create and edit geometries to be analyzed, to set up analysis runs, and to plot the results from analyzed runs. The second program is the Unified Distributed Panel (UDP) program, which is used to perform subsonic and supersonic analysis of geometries. For subsonic Mach numbers, UDP uses a combination of slender-body theory, linear source and vortex paneling methods, and empirical base drag analysis. For supersonic Mach numbers, UDP adds in the effects from an empirical wave drag analysis. The third program is the Hypersonic Arbitrary Body Program (HABP), and is used to perform analysis of the hypersonic aerodynamic characteristics of geometries. HABP uses hypersonic impact methods for its analysis. Both HABP and UDP use empirical skin friction methods to calculate skin friction forces. HABP is a modified version of the Gentry code developed in the 1960's by McDonnell Douglas, and UDP and HABP were developed in the 1970's and 1980's, respectively, by Rockwell International.¹

All three of these programs have been periodically updated to enhance their capabilities and enable them to be run on different computer systems. At one time or another, each has been run on an IBM mainframe, and Sun and VAX workstations. Currently, the Vehicle Analysis Branch runs APAS on Silicon Graphics (SGI) machines. All of the computers running APAS have had a UNIX-based operating system, and all of them have supported the Tektronics Plot10 graphics library. The Plot10 library was used by the creators of APAS to support the graphical requirements of APAS itself. Primarily, these requirements were to be able to draw and interactively edit the geometry components to be analyzed, and to display data from the analysis performed by UDP and HABP. (UDP and HABP themselves do not require any graphical support, and hence do not make use of the Plot10 library.)

Over the last few years, it has been increasingly difficult to find emulation programs that can support the interactive use of the Tektronics Plot10 library. As a result, there is a movement underway to replace the Plot10 references in APAS with a more modern standard X-based UNIX graphics package.

In addition to the increasing difficulty of finding proper emulation, the last decade has seen an explosion in the use of personal computers (PC's). There are now orders of magnitude more people who have access to PC's than to large, expensive UNIX mainframes and workstations. Accordingly, there has been an increasing call for a version of APAS that will run on a PC. This project was undertaken to fulfill that call. It was the aim of this research project to create a version of APAS system to run on a late-model IBM PC compatible, to validate the solutions given by this new version against the existing (UNIX) version, and to package this version in a form that allows the easy distribution of both the executable files and the source codes.

Approach Taken

The approach taken to this project involved a number of steps. First, the source files for APAS, UDP, and HABP were gathered and transferred over to an IBM PC compatible computer. Then, time was spent with each of the codes getting them to compile properly. At this stage it was necessary to change some calls to nonstandard FORTRAN-77 routines. Once each program was compiled and linked, APAS itself was debugged to a rough but usable level; bugs which would crash the program during normal execution were fixed, but less serious problems were left alone. It should be noted at this point that a contractor had previously written a set of routines to mimic the Plot10 library on a PC using the Watcom FORTRAN 77 compiler's graphics library. However, these routines had never been fully integrated into APAS or tested with it. It was at this point that the testing and debugging of these new Plot10-substitute routines occurred.

Once APAS had reached a level where geometries could be created and set up for analysis, the process of testing and debugging HABP and UDP began. At first, very simple geometries were created and analyzed to catch any immediate run-time errors that occurred. Then, more complex configurations such as a model of the Space Shuttle Orbiter were used for direct comparison with results obtained from the SGI versions of HABP and UDP. During this process a number a bugs were discovered, some of which were the result of errors in the original source code. These errors may need to be corrected in the current versions of the codes. This process continued until the PC and SGI versions of HABP and UDP showed identical results on multiple configurations.

Once HABP and UDP were deemed to be operating properly, work on APAS itself was finished. A number of minor, mostly cosmetic bugs were corrected. In addition, new code was written to allow the use of a mouse when using the Plot10-substitute routines. (The original APAS allows the use of the mouse.) Then, once APAS was finished, a few external utilities that typically accompany the APAS system were compiled and tested. A listing of all of the changes made to each of the APAS programs, and the reasons why the changes were made was then documented.

The final step in this project was to create a set of compressed files for easy distribution of APAS. One compressed file contains the executable files and other files

necessary to run APAS. This file also contains a README text file to explain how to run each program, as well as and other information about the PC variations from the UNIX APAS. The other compressed file contains the source codes and information about what was changed in the source files.

LaRC Equipment Used

Only a small amount of equipment was needed for this project. The primary piece of equipment used was a Dell 66 MHz Pentium computer with the Watcom FORTRAN 77 32 bit Compiler (version 9.5c) in the Vehicle Analysis Branch. The Silicon Graphics computers in the branch were used to provide the benchmark platform against which the PC version of APAS was compared. The Langley Technical Library was also used for references about programming for the mouse on a PC.

Results

Two different topics will be addressed in this section. The first will be a description of the types of corrections which needed to be made to the original APAS source codes, and what action was taken to correct them. The second will be a brief comparison of the data obtained from APAS with data from the Aerodynamic Design Data Book for the Space Shuttle Orbiter.

Problems encountered in conversion

A detailed discussion of every change made to the APAS source files is beyond the scope of this report. Instead, a general discussion of the types of problems encountered and what was done to resolve them will be included here.

The first major type of problem encountered in converting APAS was the occurrence of compiler- or platform-specific references in the original source codes. A number of these problems were quite simple, such as the use of double quotes in the INCLUDE statements in UDP, which had to be replaced by single quotes for the Watcom system. Some of these problems were caused by calls to non-standard FORTRAN 77 functions that perform system-specific tasks such as accessing the current system time or date, reading additional words used on the command line when the executable file was invoked, and exiting the program with a return code. They were fixed by replacing the SGI compiler function calls with their appropriate Watcom compiler equivalents. In addition, a number of calls to functions which did not have Watcom equivalents were encountered. These were primarily trigonometric functions using degrees instead of radians. The prevalence of these functions in the source codes made it simpler to create a few additional functions with a radians to degrees conversion than it would have been to replace all of the calls to them in the source files. A final change in this category in the handling of files. On previous systems where APAS was used, the limits on the number of files open and the numbers of characters in a filename were much larger than on the PC. It was necessary to add a statement to HABP and UDP which increased the number

of files that could be opened by the programs, and to truncate any filenames that were too long.

For the most part, these types of problems were relatively quick and easy to solve (although INCLUDE statements with double quotes appeared in about three fourths of the UDP subroutines, and were time-consuming to replace). Once aware of them, they did not present much difficulty to correct. However, the use of compiler-specific routines in the source codes will continue to present a barrier when taking the software from one platform or compiler to another.

The second major class of problems encountered was errors in the original source codes. A few of these errors, such as incorrect FORMAT statements, were easy to spot and relatively harmless, but most caused substantial problems and were hard to find. Most of the errors centered around the definition, typing, and initialization of variables. In a few cases, it was obvious that an INCLUDE file containing a needed common block was missing. In these cases, the required INCLUDE statements were added. In a larger number of cases, an apparent typographical error led to using a variable that was never defined or initialized in a particular routine. Most of these cases were remedied by replacing the variable with another variable in the routine which seemed like the one that original programmers must have intended. A number of errors were also caused by differences in the arguments of COMMON blocks in different routines, or by assigning variables different data types as they are passed from one routine to the next. These errors were corrected when they seemed to cause a problem. Lastly, a couple of coding errors were discovered that resulted when the code was prepared for use with the SGI compiler. Apparently, the SGI compiler cannot handle using nested exponents on a single line. In a couple of instances in HABP, the lines replacing the original nested exponentiation were mathematically different from the originals, and hence generated incorrect results. These lines were changed to restore the original mathematical expressions.

This class of problems was far more insidious than the first class and occupied a large portion of the time spent testing and debugging the APAS programs. The instances of errors that seemed to be causing problems appear to have been fixed. But there is no guarantee that errors of this type will not resurface. There are still quite a number of common blocks that have multiple type definitions, and quite a number of variables with multiple types, that were left in place. The potential exists for many more errors of this type to surface in the future if the code is moved to other systems or if more thorough testing is done.

The last major type of problem encountered when converting APAS to run on a PC was the difference in graphical formats used in the PC and Plot10 systems. These problems were only encountered in APAS, since HABP and UDP do not have any graphics. A number of problems were caused by what appeared to be errors in the Plot10-substitute library code. They were most likely the result of not fully integrating and testing the new library with APAS, and all of them that were found were fixed. In a few cases, minor errors in text or symbol positions had to be corrected. They were probably caused by the differences in a PC screen resolution, which is 1024x768 pixels and the Tektronics terminal, which was 4096x3180 pixels. A final problem of this type was the difference in the use of graphical and text-based text between the two systems. On a Tektronics terminal, both graphically drawn text and regular text could be positioned in

the same fashion. The PC uses a different means to position each kind of text. In cases where the text position was considered critical, graphical text reads and writes were substituted for text-based ones in the PC version.

Most of these types of issues were not absolutely critical to the functioning of APAS, but correcting them made APAS a much nicer program to use. The most serious problems of this type were actually errors in the original source code (such as multiply-typed variables) which did not cause problems for Plot10 graphics, but could cause the Watcom PC graphics to crash the program.

One additional issue that needed to be addressed in this area was the use of a mouse to move the cursor when viewing or editing the geometry components of a vehicle. The original Plot10 library supported the use of the mouse, since it was directly supported by the keyboard interface. (Originally, before the use of the mouse became prevalent, keyboard thumbwheels were used.) The Plot10 replacement routines for the PC did not include mouse support, instead relying on the keyboard cursor keys to move the cursor around the screen. While this performs adequately, it was desirable to have mouse support on the PC as well in order to more closely copy the "feel" of the original APAS. To do this on a PC, a few additional lines of code had to be written to issue calls to DOS assembly language interrupts to position and reposition the mouse cursor. Unfortunately, the resulting mouse movement seemed jerky and was hard to position precisely. As a result, a dual-movement method was used. The user can move the cursor with the mouse and then press the mouse button, which will allow the user to either reposition the cursor with the cursor keys or to enter a desired keyboard command.

APAS comparison for Space Shuttle Orbiter configuration

Comparisons of the results obtained from APAS analysis of the longitudinal aerodynamic characteristics of the Space Shuttle Orbiter with the values contained in the Shuttle's Aerodynamic Design Data Book (ADDB) will now be discussed. The comparisons have been made for varying angles of attack at two Mach numbers, one subsonic (Mach = 0.6) and one hypersonic (Mach = 15). It should be noted that many more comparisons and validations of the code than those discussed here were made. However, in the interest of space, only these two cases will be discussed.

The APAS model of the Shuttle Orbiter used for this comparison is shown in Figure 1. The data for the subsonic comparison is shown in Figure 2. The values of the lift, drag, and pitching moment coefficients (C_l , C_D , and C_m) are shown as functions of angle of attack (α). The results obtained from the APAS (UDP) analysis are shown as straight lines, and values from the Aerodynamic Design Data Book are shown as "+" characters. From Figure 2 it can be seen that the lift and drag curves are generally in fairly good agreement with the values from the ADDB. UDP slightly overpredicted the slope of the lift curve and slightly underpredicted the drag, but these differences are small. The pitching moment curve, however is significantly different from the values in the ADDB. This is probably due to the fact that only linear terms were used in this particular analysis. Other studies done with APAS have shown considerably better agreement.

The data for the hypersonic comparison is shown in Figure 3. As in Figure 2, the values of the lift, drag, and pitching moment coefficients (C_l , C_D , and C_m) are shown as

functions of angle of attack (α), and the results obtained from APAS (HABP) are shown as straight lines. The lift and drag curves show even better agreement with the ADDB data for this case than they did for the subsonic case. The pitching moment curve shows somewhat better agreement in this case than in the subsonic case, although it is largely in error in some places. This is most likely due to an error in the prediction in the center of pressure in HABP. For the shuttle, this error has been shown to be about two percent of the body length. If the center of gravity in APAS were to be moved forward by two percent, it is likely that the discrepancy between the APAS data and the data from the ADDB would be substantially less.^{2,3}

Conclusions

The Aerodynamic Preliminary Analysis System (APAS) software has been converted from a Silicon Graphics UNIX-based platform to run on a DOS-based IBM PC-compatible. The APAS software has been tested, debugged and validated for a number of representative cases. APAS has now been packaged in a way that it is easy to distribute both the executable files and the source files, and in such a way that it is easy for whoever receives the distribution files to install and run them. A discussion of the relevant background information, the steps taken to accomplish the conversion and the types of problems encountered during the conversion has been made. A brief comparison of data obtained using APAS with data from another source has also been made. The following conclusions are drawn:

1) The types of problems encountered in converting APAS to a run on a PC included those caused by compiler-specific references, those caused by errors in the original source codes, and those due to differences between the PC and Tektronics Plot10 graphics formats.

2) A number of errors in the original source codes still exist. Care must be taken to insure that these errors do not cause problems in the future.

References

1. Engelund, W. C. and Ware, G. M. *Aerodynamic Predictions and Experimental Results for an Advanced Manned Launch System Orbiter Configuration*, AIAA Paper 93-1020, February 1993.

2. *Aerodynamic Design Data Book Vol. I: Shuttle Orbiter Vehicle*, Rockwell International, October 1978.

3. Cruz, C. I. and Engelund, W. C. *An Aerodynamic Preliminary Analysis System (APAS) Calibration Report: Space Shuttle Orbiter*. Paper presented at The Society of Hispanic Professional Engineers 6th Annual Eastern Technical and Career Conference, November, 1992.

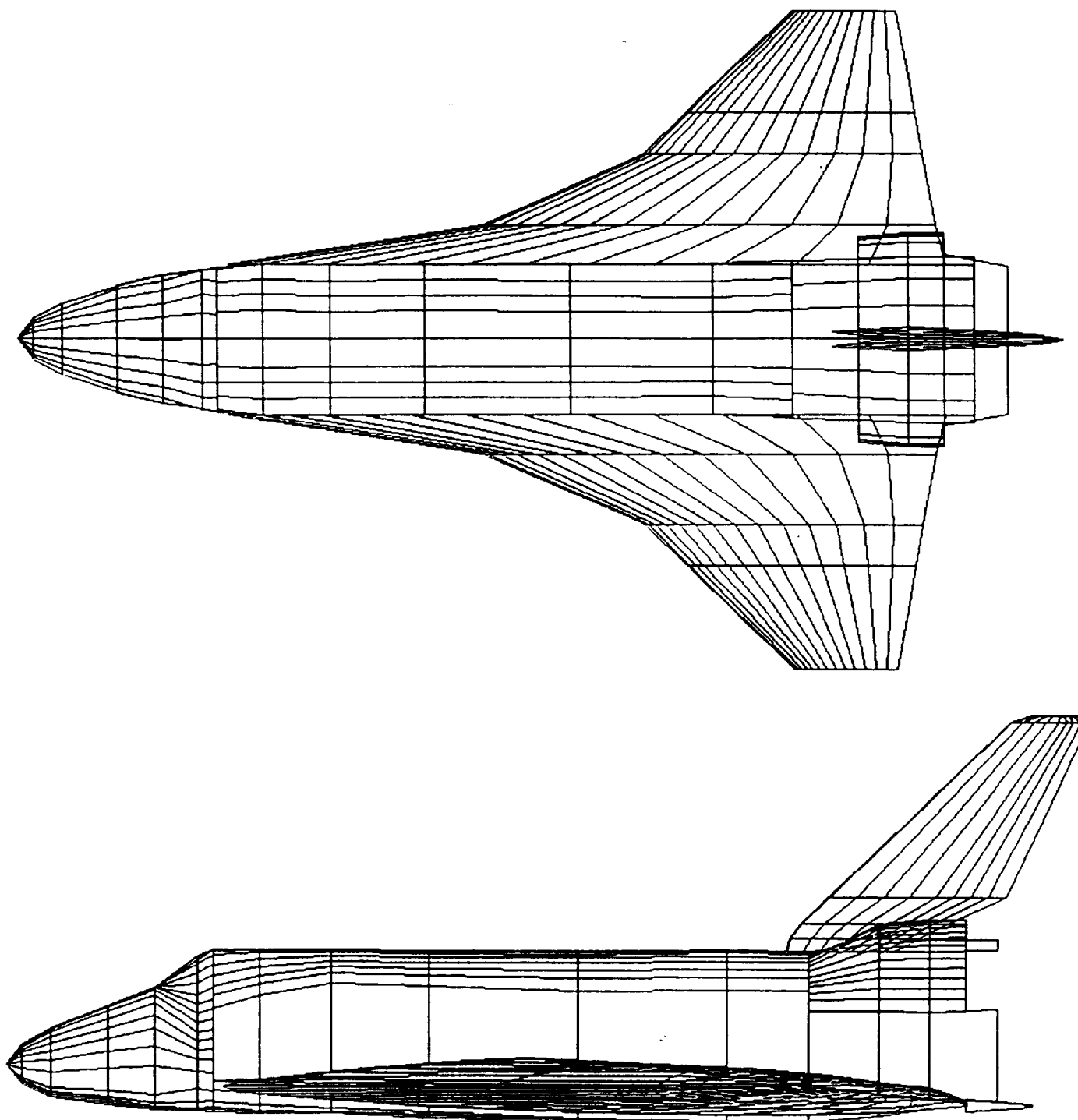


Figure 1. APAS model of Space Shuttle Orbiter, top view (top) and side view (bottom).

Subsonic Longitudinal Data

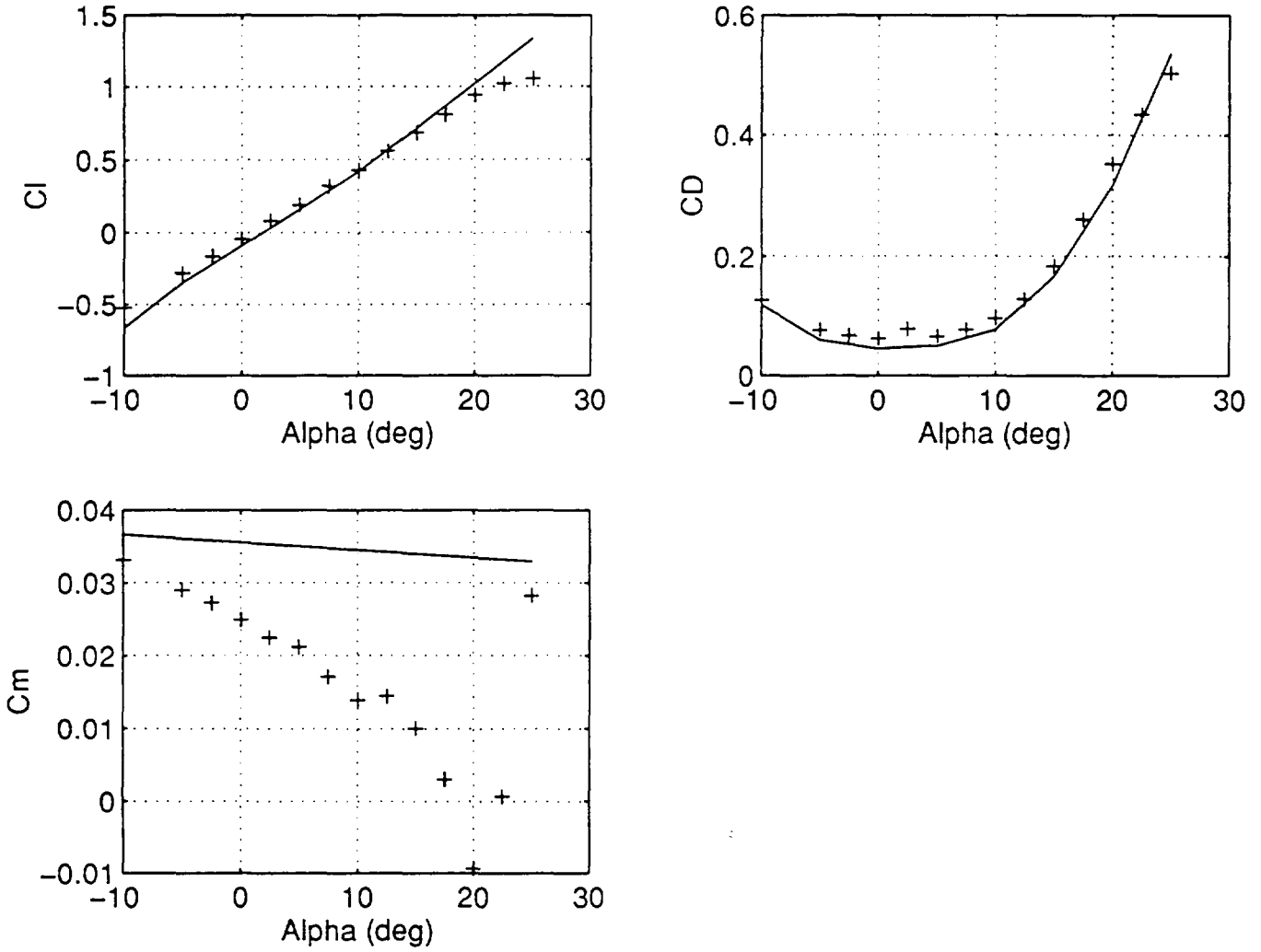


Figure 2. Subsonic longitudinal aerodynamic data.

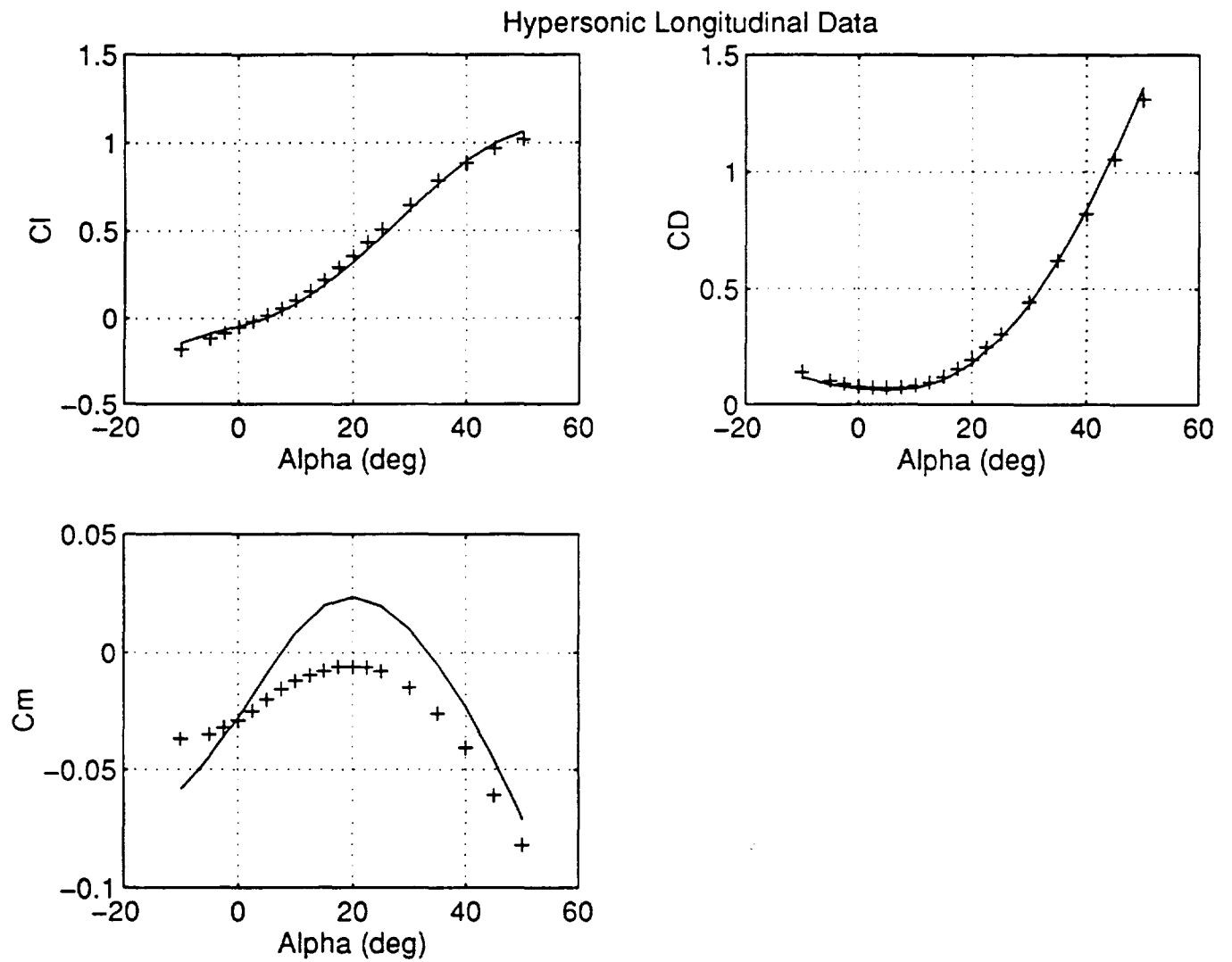


Figure 3. Hypersonic longitudinal aerodynamic data.