N72-121

NASA TECHNICAL MEMORANDUM

NASA TM X-67968

NASA TM X-67968

GRAPHIC DISPLAYS FOR LARGE AERODYNAMIC TEST FACILITIES

by Daniel L. Whipple
Lewis Research Center
Cleveland, Ohio
November, 1971

GRAPHIC DISPLAYS FOR LARGE AERODYNAMIC TEST FACILITIES

by Daniel L. Whipple

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio

E-6474

ABSTRACT

A combination of software and hardware for a time-sharing computer is described which allows the user to obtain an on-line data display in the control room of a large research facility. Display and hard copy of alphanumeric data as well as graphical data can be obtained as desired by the user. In addition, a number of utility programs provide for on-line graphic editing, program control, data manipulation, and off-line microfilm processing.

GRAPHIC DISPLAYS FOR LARGE AERODYNAMIC TEST FACILITIES

by Daniel L. Whipple

Lewis Research Center

SUMMARY

A combination of software and hardware for a time-sharing computer is described which allows the user to obtain on-line data displays in the control room of a large research facility. Display and hard copy of alphanumeric data as well as graphical data can be obtained as desired by the user. In addition, a number of utility programs provide for on-line graphic editing, program control, data manipulation, and off-line microfilm processing.

INTRODUCTION

The effective and efficient use of a large research facility requires that a maximum of useful data be obtained in minimum time. Where an exactly predefined sequence of test points cannot be determined, it is important that current data be quickly analyzed to provide information for the test engineer. This report describes a currently operating system that provides on-line, interactive graphical data presentation for the test engineer. The results were applied in the 10- by 10-foot supersonic wind tunnel of the Lewis Research Center.

The system is designed to allow the test engineer to control its characteristics rather than forcing him to operate in a rigid predefined environment. Various output devices are accommodated, and equipment is available for interactive graphical analysis after a test run for further data reduction or preparation of final data plots.

Continuous-operation facilities have traditionally required some form of on-line analysis in order to ascertain the data points necessary during a run for efficient facility utilization. The first example (old, but still in use) is the slide rule or calculator which is used to convert observed readings into engineering units.

If permanency were desired, the data would be hand copied into a notebook or photographed from readouts. A later system (ref. 1) had the advantage of automatically recording data, but presented all data on a scale of -1000 to 1000 counts, so that the data still had to be reduced to obtain engineering units. With the advent of the large time-shared computer it became feasible to equip each control room with a typewriter terminal which, with the appropriate programs, provided output in engineering units for the test engineer (ref. 2).

The problem still remained of efficiently handling large amounts of data and presenting them in graphical form such as pressure profiles and performance comparisons. For these data the system to be described was developed.

HARDWARE

Various configurations of equipment and software were investigated. The time-shared terminal in the facility control room was augmented with a digital incremental plotter and a closed-circuit television system to allow viewing of one of two large cathode ray tube (CRT) displays located in the central computing center. An initial software (program) package was prepared, and this equipment was run during various test programs to determine what changes or additions to the hardware and software were needed. As data were taken only at discrete operating conditions of the test model, the displays presented data only as recent as the latest scan of the instrumentation rather than continuously updated data. Dynamic (high-frequency) data were not handled by this system.

After about 8 months of experimentation and use, the most useful hardware configuration for the facility, shown in figure 1, was attained. There are numerous possible equipment combinations which can be selected by the test engineer before a run. They are

(1) Typewriter terminal and keyboard, which give output in orderly format and engineering units.

(2) Typewriter and digital plotter, which give hard copies of plots as selected from the keyboard

(3) Typewriter, digital plotter, and TV/CRT display, which give quasi-real-time graphical displays on the TV under user control with the option of providing a hard copy of any graph upon request

In all of these combinations the user has the option of selecting plots to be automatically microfilmed off line.

During the development period, the following problem areas were encountered:

(1) Hard copies of plots and alphanumeric data took excessive time to be produced on the digital plotter.

(2) The TV/CRT system suffered from a lack of resolution.

(3) The CRT could not conveniently produce temporary alphanumeric listings to replace the typewriter terminal.

(4) CRT refresh buffer memory size was inadequate.

These problems can be resolved by obtaining a storage tube type display and using high-speed data communications. These approaches should provide the speed and resolution needed and eliminate the buffer size problems. The display also has an alphanumeric mode, which allows it to be used in place of the typewriter. While the screen size is somewhat smaller than the existing graphics displays, it is more than adequate for control-room use.

SOFTWARE

The software needed to control and drive the hardware was the most difficult part of this system to develop for the following reasons:

(1) None existed in the form needed (a device-independent package).

(2) No knowledge existed as to exactly how the system should be used or what user services should be provided.

(3) Various types of equipment had to be used together that had not previously been interconnected on the time-sharing system.

(4) Graphical data displayed or plotted on two different devices should look as much the same as possible.

(5) The control-room users of the system were not experienced programmers.

The initial software routines were custom designed programs, using the existing device-dependent software. These were applicable to only one test program and were lengthy and difficult to code. They did, however, allow valuable experience to be gained. Further testing revealed the following needs:

(1) Temporary storage of data from many scans for composite performance and comparison plots, since data were recorded and made available one data scan at a time

(2) An on-line graphic editing capability of modifying any aspect of the original plot to produce a more finished product without requiring user program changes

(3) Very simple user programming, since most users had neither the time nor the experience to do extensive software coding

(4) A generalized program to control (by user commands) the sequence and timing of plot presentation

The currently operating software package meets all of these requirements and also provides some utility programs for the user to call as needed. This package currently allows for linear, logarithmic, and polar plots under user control along with a conversational graphic editor. The master control program allows the user to select his hardware configuration and then makes available to him all of his predefined plots as well as all of the graphics package services. The capability is provided to store a part of the current set of data in the graphics system for future recall and use without going through the main data accession routines.

A typical plot, from user selection to display on the CRT takes about 200 to 300 milliseconds of computer time. A hard copy in the control room takes about the same computer time but much more actual elapsed time (2 to 10 min), since the digital plotter is a low-speed mechanical device. Improved hardware should reduce this time to less than 20 seconds. Total elapsed time for a CRT display plot from selection to completion is 2 to 30 seconds, depending on the time-sharing system loading.

PROGRAM DETAILS

The central focal point of the software is an area of storage called PLTDTA which provides a place to store all of the information needed to generate one plot (see the appendix). This storage area is divided into two arrays of numbers; one floating point and the other integer (compatable with FORTRAN conventions). These arrays contain the basic plot information as well as the data and are independent of the type of output device. The current size of PLTDTA was determined solely by the total number of data points for any one plot, which is 500 points, or up to 10 curves of 50 points each. In addition, space is provided for up to 200 characters of legend text and all required plot characteristic information.

The graphical subroutines divide into three general groups:

(1) Those that write into PLTDTA

(2) Those that read from PLTDTA

(3) Those that modify information already in PLTDTA

The first group, those that write into PLTDTA, are the ones most generally used by the test engineer. Two means are provided (subroutine calls or word by word access) to place data and graphical control information into PLTDTA. This is usually under the control of a driving program which offers the user a choice of branching directions. After the user responds, the program goes to the user-written subroutines for the desired plot description, fills PLTDTA, and then regains control when the array is filled.

The array needs to be filled with enough information for one plot. Thus, only the following are required:

(1) Minimum and maximum values of user's units

(2) Plot size scaling factor

(3) Plot grid location and size in normalized coordinates (0.0 to 10.0 for all display devices)

(4) Grid type (linear, log, or polar)

(5) Number of curves (0 to 10), number of legends (0 to 10), and number of points for each curve

This information would produce a basic plot. Additional information is generally given to create a more readable and useful plot. All data are in user's units (the range of the data) except for a few absolute positions (such as grid location). All the output devices appear to the user as having a normalized size of 0.0 to 10.0 units in both the X and the Y directions.

The second group reads in information from PLTDTA and generates the plot. The user selects the type of device he wants to use and calls the appropriate display routine by giving the control program the proper response. These programs automatically set up the plotting package for the desired device and start the plot. The program which does the actual plot generation uses the data and control information from PLTDTA to output the plot through the driving routines for the various output devices.

After the plot is completed, control is returned to the user for another branching direction. Each time a plot generation routine is called, it generates a plot on the selected device from the current information in PLTDTA.

The third group only modifies the information currently in PLTDTA. It consists mostly of the graphical editing program. This is a subroutine which the user can call to change dynamically a plot currently in PLTDTA. It allows the user to use commands and arguments which temporarily modify any of the plot characteristics without requiring the user to change his plotting program. In addition, the plot information can be displayed at the terminal, and the user can manipulate the data either one point at a time or in logical groups. This is the means by which a user can clean up a plot and put it in a desired format without changing his own plot subroutines.

There are also additional subroutines which combine the functions just described to allow a user to create an entire plot from keyboard commands and will even prompt him for the information needed to construct the plot.

It should be mentioned that all user-written programs were in FORTRAN IV, as was the majority of the graphics package. The 10 percent or so of the package written in BAL (assembler language) was written that way to make the use of the package easier and more efficient for the test engineer. All the programs written in BAL can be and were originally written in FORTRAN IV.

Figures 2 to 7 give examples of plots made with this graphics package. Figure 2 is a linear grid with full grid lines, two data curves, and two legends. Figure 3 is the same plot as shown in figure 2 with the grid code changed so that tick marks are generated instead of the full grid. Figure 4 is also the same plot, but with the ordinate reduced to about 0.6 of the value used in figures 2 and 3. Figures 5 and 6 are combinations of linear and logarithmic grids, such as those used for frequency response data. Figure 7 is a polar plot presented with no connecting lines between data points.

SUMMARY OF RESULTS

A display system was assembled which utilizes a time-shared computer to give on-line data displays in the control room of a large research facility. New hardware will provide increased speed and ease of operation, while improving the quality of the displayed and printed plots. The software package provides for simple user programming, dynamic plot editing, and generalized control and sequencing of data display. The software requires the facility test engineer to write one subroutine describing the desired graphs in a device-independent format. He may then utilize on-line data display and manipulation for viewing data on a variety of devices and for generating plots usable in research reports and other presentations.

APPENDIX - DESCRIPTION OF PLOT ARRAY[1]

Z Array Requirements
Z(111, 10)

Z (1, 1) → Z (100, 10)	1000 words (4-byte words) for 500 points for X and Y (or R and θ) plot data. May be used as 10 curves at 50 points each; 5 curves of 100 points each; 1 curve of 500 points; etc.
Z (101, 1) → Z (105, 10)	10 sets of 20-byte areas to store a legend or title of up to 20 characters.
Z (106, 1) → Z (106, 10)	X position of legend (0. 0 to 10. 0).
Z (107, 1) → Z (107, 10)	Y position of legend (0. 0 to 10. 0).
*Z (108, 1)	Minimum X (or R) value in user's units.
*Z (108, 2)	Maximum X (or R) value in user's units.
*Z (108, 3)	Minimum Y (or θ) value in user's units.
*Z (108, 4)	Maximum Y (or θ) value in user's units.
Z (108, 5)	X direction (or R) grid increment in user's units. If zero or negative, no grid lines at all. Not used for log plots but must still have nonzero value.
Z (108, 6)	Y direction (or θ) grid increment in user's units. Same rules at Z (108, 5).
*Z (108, 7)	Size factor, normally 1. 0; if any other value, will scale entire plot and all alphanumerics by that amount.
Z (108, 8)	4 bytes for a 4-character reading number to be put at XLOC = 0. 5, YLOC = 0. 5 on the plot.
Z (109, 1) → Z (109, 10)	Ten spaces for alphanumeric plot characters for the 10 data curves - the characters must be left justified in these 4 bytes.
*Z (110, 1)	XLO (or X origin) of the plot.
*Z (110, 2)	XHI (or Y origin) of the plot.
*Z (110, 3)	YLO (or R total) of the plot.
*Z (110, 4)	YHI (or R offset) of the plot.

All other Z elements are spares.

---

[1]Asterisk indicates that some value must be given for practical results.

JZ Array Elements
JZ(8, 10)

JZ (1, 1) → JZ (1, 10)	Direction code for legends 1 to 10: 0 for horizontal, 1 for vertical.
JZ (2, 1) → JZ (2, 10)	Character size for legends 1 to 10; must be 1, 2, 3, or 4.
JZ (3, 1) → JZ (3, 10)	Character size for plots 1 to 10. Same sizes as legends; will use Z (109, N) alphanumeric character. If this is negative, a special character will be plotted; if -1 to -4, no lines will join the symbols; if -5 to -8; straight lines will join the symbols.
JZ (4, 1) → JZ (4, 10)	Special plot character for plots 1 to 10. Denoted by integer value 0 to 255 (normally 0 to 13). This is used only if JZ (3, N) is negative.
JZ (5, 1) → JZ (5, 10)	Number of data points for curves 1 to 10. Example: If curve 3 had 27 data points, JZ (5, 3) would be set to 27. If it were zero, no data points would be plotted.
*JZ (6, 5) → JZ (6, 6)	X and Y (or R and θ) grid codes as follows: 1, linear tick marks; 2, linear lines; 3, log tick marks; 4, log lines; 5, polar plot.
JZ (6, 7) → JZ (6, 8)	X and Y (or R and θ) number of labels on grid lines. If 0 or 1, no labels will be drawn. For log plots this should equal number of cycles +1.
JZ (6, 9) → JZ (7, 9)	X and Y (or R and θ) label character sizes. May have values 1 to 4 (see JZ (2, N)).
JZ (6, 10) → JZ (7, 10)	X and Y (or R and θ) label significant figures as follows: >1, that number of digits after the decimal point; -1, only the integer portion of the number; <1, /N/ - 1 digits are cut from the integer value.
*JZ (7, 1)	Number of legends; must equal 0 to 10.
*JZ (7, 2)	Number of curves of data; must equal 0 to 10.
*JZ (7, 4)	Switch for reading number (Z (108, 8)); if 0, it is not plotted; if 1, it is plotted.
JZ (7, 5)	Number of cycles for log grid in the X direction - used only if JZ (6, 6) = 3 or 4.
JZ (7, 6)	Number of cycles for log grid in the Y direction.

REFERENCES

1. Staff of the Lewis Laboratory: Central Automatic Data Processing System. NACA TN 4212, 1958.

2. Putt, Charles W.; and Goldberg, Frederic N.: Use of Time-Sharing Computer to Support Several Large Facilities. NASA TM X-52580, 1969.

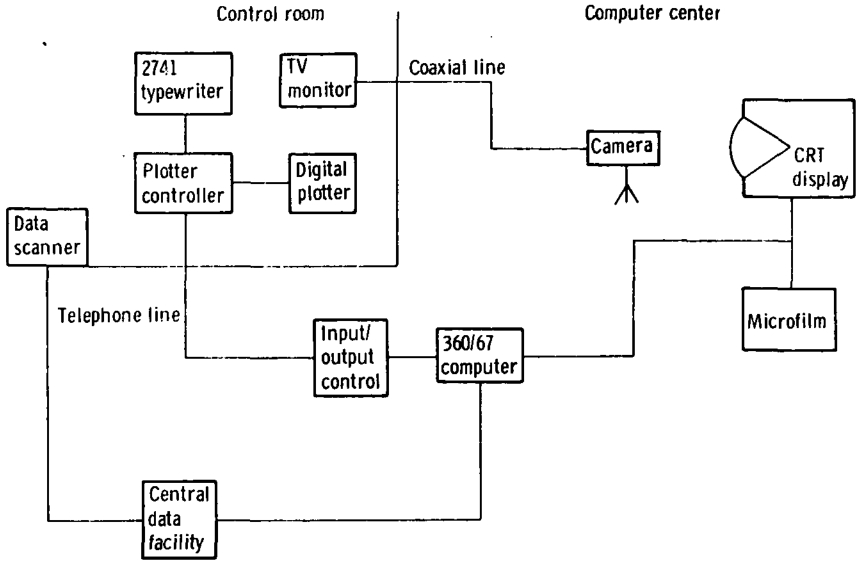

Figure 1. - Diagram of hardware connections.

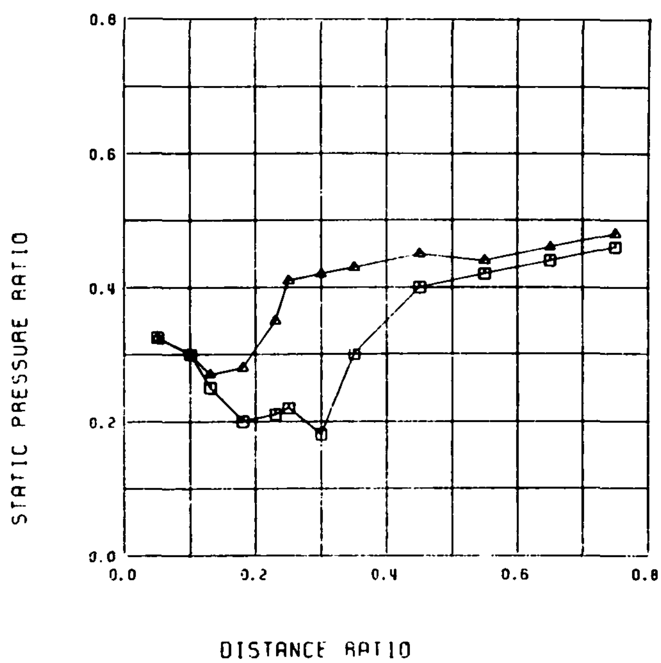

Figure 2. - Plot with linear grid lines.

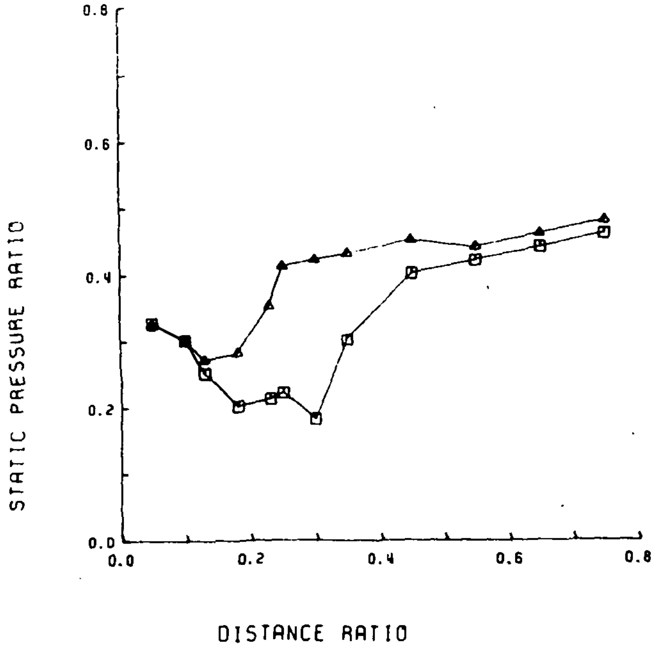

Figure 3. - Plot with linear ticks.

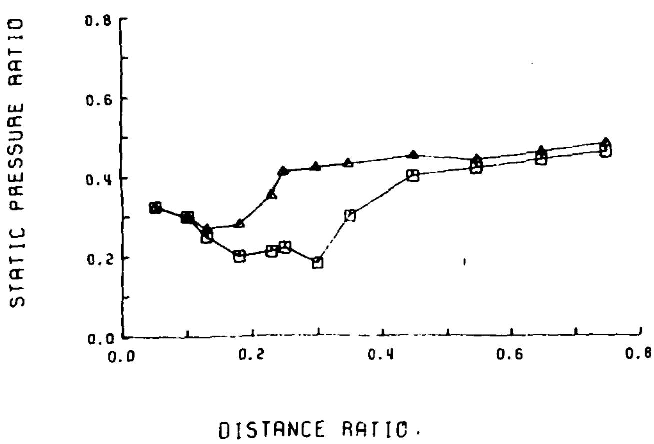

Figure 4. - Plot with linear ticks and reduced ordinate scale.

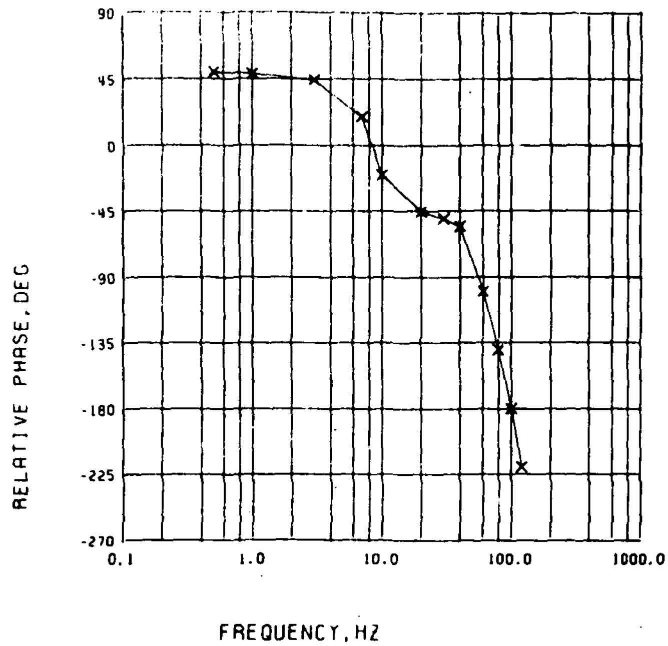

Figure 5. - Plot with semilog grid.

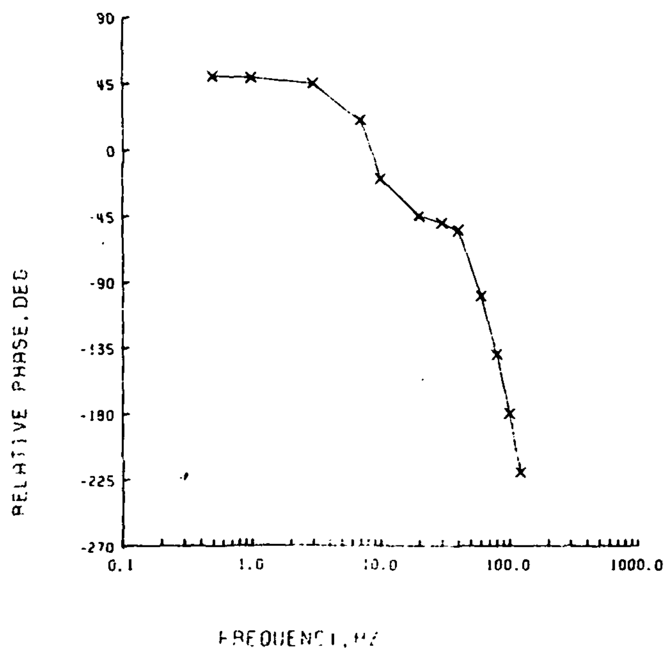

Figure 6. - Plot with semilog ticks.

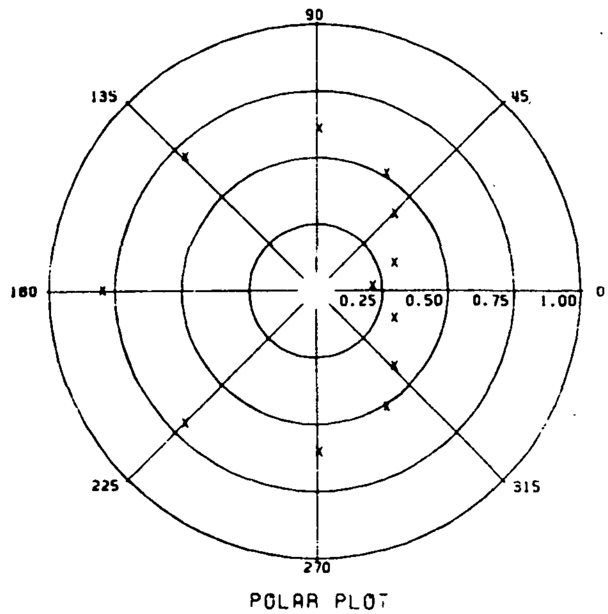

Figure 7. - Polar plot.