

NASA Technical Memorandum 107128  
AIAA-96-0049

NASA-TM-107128

19960011373

# X Based Interactive Computer Graphics Applications for Aerodynamic Design and Education

Thomas J. Benson  
*Lewis Research Center  
Cleveland, Ohio*

and

C. Fred Higgs III  
*Rensselaer Polytechnic Institute  
Troy, New York*

Prepared for the  
34th Aerospace Sciences Meeting and Exhibit  
sponsored by the American Institute of Aeronautics and Astronautics  
Reno, Nevada, January 15-18, 1995



National Aeronautics and  
Space Administration

LIBRARY COPY

JUN 23 1995

LANGLEY RESEARCH CENTER  
LIBRARY NASA  
HAMPTON, VIRGINIA



# X Based Interactive Computer Graphics Applications for Aerodynamic Design and Education

Thomas J. Benson\*

NASA Lewis Research Center, Cleveland, Ohio 44135

and

C. Fred Higgs III†

Rensselaer Polytechnic Institute, Troy, NY 12180

November 28, 1995

## ABSTRACT

Six computer applications packages have been developed to solve a variety of aerodynamic problems in an interactive environment on a single workstation. The packages perform classical one dimensional analyses under the control of a graphical user interface and can be used for preliminary design or educational purposes. The programs were originally developed on a Silicon Graphics workstation and used the GL version of the FORMS library as the graphical user interface. These programs have recently been converted to the XFORMS library of X based graphics widgets and have been tested on SGI, IBM, Sun, HP, and PC-Linux computers. The paper will show results from the new packages, detailing the change in design and philosophy brought about by the move to X based graphics using the new VU-DUCT program as a prime example. VU-DUCT has been developed as an educational package for the study of subsonic open and closed loop wind tunnels.

was conducted using charts, tables and graphs of the performance of earlier similar configurations and final design was tested and verified using wind tunnel results. With the advent of large, powerful mainframe computers, some of the preliminary design tables and graphs could be numerically generated by solving the equations of motion and some of the final design results could be verified and studied using computational fluid dynamics (CFD). Today's workstations and personal computers have computing power equal to that of the older mainframes and when coupled with a window operating system and a graphical user interface, (GUI), can now be used to develop preliminary designs interactively. The software developed for preliminary aerodynamic design can also be used for educational purposes to study the performance of various aircraft components. In the propulsion area, several computer applications have recently been developed for education and preliminary design, Refs. 1 through 4.

## INTRODUCTION

Recent advances in computer related technologies are changing the ways that engineers solve aerodynamic problems. In the past, preliminary design

This paper will present extensions of work begun by the author in 1994, Refs. 5 through 8, concerning the development of a family of computer applications to perform preliminary design and educational analysis using interactive computer graphics on a single workstation. In Ref. 5 an interactive inlet design tool was developed to solve for the flow through external compression inlets. Through the use of a GUI, the designer can change the geometry, and the upstream and downstream flow conditions and immediately see the effects on inlet performance and drag. As the geometry and flow conditions are altered, the application recomputes the

\*Senior Research Engineer, Member AIAA

†Graduate Student

© Copyright © 1994 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

important flow variables and redisplay the geometry, shock wave locations, and output flow parameters. In Ref. 6 some of the coding from the the inlet design program was used to produce a compressible flow simulator for undergraduate study. This simulator computes and displays the inviscid flow field created by supersonic flow past a single wedge, two opposing wedges, or two wedges in succession. Interactive features were added to this package to allow the student to record and plot data produced by the simulator. To verify the results from the first two simulators, a series of compressible flow calculators were produced and reported in Ref. 7. While in Ref. 8, some of the ideas developed in the educational simulator were used to develop an educational package to study simple turbojet engines. All of these packages were developed on a Silicon Graphics workstation using GL graphics libraries and the FORMS library of widgets. As such, the applications could only be run on SGI workstations.

All of the applications packages have been expanded to solve additional flow problems during the last year and two new packages have been developed. The inlet design tool can now calculate mixed compression inlets, while the turbojet package can now analyze turbojets with afterburners and two spool turbofan engines. One of the new packages, called VU-DUCT, can be used to design or study the flow in subsonic open or closed loop wind tunnels. The other, called DUCTSIZER, is used to design or study the flow through multiply connected air conditioning ducts. There are plans to improve, upgrade, and implement additional features in all the packages.

The most significant change in the past year has involved the conversion of all the packages from GL based graphics to X based graphics. The original intent was to make the packages available to more users since X graphics has become a standard on Unix workstations and some PC's. The X based widget set XFORMS which is used in the simulators has been made available for IBM, SGI, HP, Sun, DEC, Cray and PC-Linux machines. Therefore, any user having access to one of these types of machines can now operate the applications in a stand-alone mode. However, in distributing early versions of the converted software to universities, it became apparent that these users were more interested in exercising the full capabilities of the X

software to operate in the client/server mode. The software would be executing on a workstation but be controlled and viewed from a less expensive PC running an X windows emulator. This created some problems in the software which were not encountered in the GL based version running on a single machine. The solution of these problems has led to a major redesign of the software.

## DESCRIPTION OF APPLICATIONS

A brief description of the development of these applications can aid future users or modifiers of the packages. Work began on the first of the applications in May of 1993. Because the author has access to a Silicon Graphics 4D25TG Personal Iris, the applications were originally developed on this platform and written in the C programming language to be fully compatible with the graphics. Using the new VU-DUCT simulator as an example for all six simulators, Fig. 1 shows the basic layout of the graphics display for the GL version. This

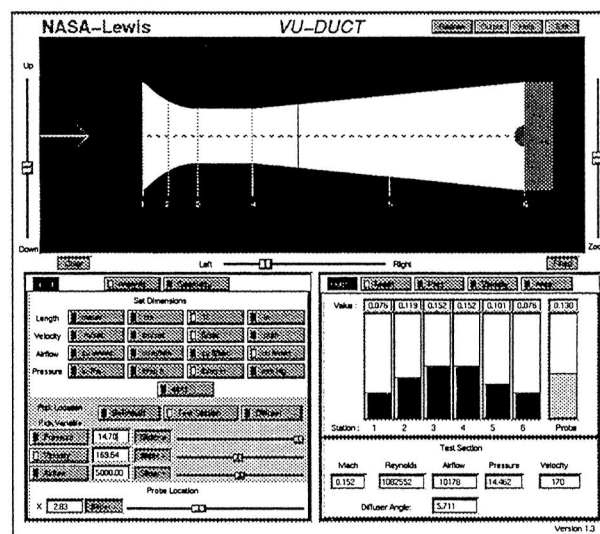


Figure 1: GL version of VU-DUCT simulator

layout begins with a background form on which numerous buttons, sliders, and other forms are placed. The upper right corner of the background form contains buttons to invoke the on-line help screens, the I/O screens, and to redraw and exit the package. The upper portion of the simulator includes a large

viewing area in which a schematic drawing appears. There are sliders located around the view window to size and locate the image within the window, and there are several buttons below the window which control the appearance of the image; line drawing or filled, black and white or colored. Below the main view window are two boxes; the one on the left controls input to the simulator while the one on the right displays the output from the simulator. There are two possible input panels on the left and the user chooses which panel to display by the lighted radio buttons at the top of the box. Each panel is itself a form containing buttons, sliders, input boxes and labels. The forms are placed by the software at a specific location on the background form and cannot be resized or relocated. The output box at the right contains two panels. The top output panel displays selected variables both numerically and as moving bar graphs. The lower output panel gives numerical values of flow variables in the test section.

It was recognized early in the development of the codes that they would be modified to use alternate graphics systems. To facilitate the transition from GL graphics, a design philosophy was adopted as shown in Fig. 2. Each of the codes are cleanly di-

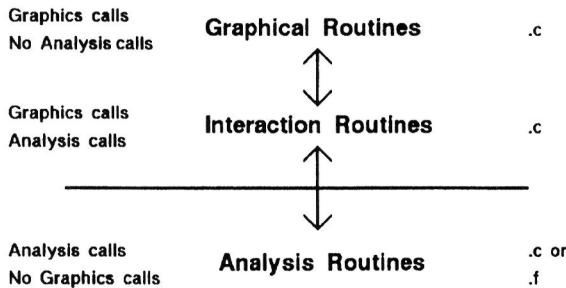


Figure 2: Philosophy of code development

vided into three sections. The first section deals only with the layout and drawing of the GUI. The second section performs the interactions between the GUI and the analysis portion of the code. The third section contains only the analysis routines for each application and is totally independent of the graphics system used on a particular platform. The graphics routines and interface routines are necessarily written in C; the analysis routines can be written in either FORTRAN or C. This program configuration leads to some inefficiencies, such as flags being set to pass information, but a large portion of

the code remains unchanged when moving from one graphics system to another.

The graphics routines consists of two parts, one to produce the schematic drawing and the other to produce the input and output widgets. The author wrote the schematic drawing portion of the codes using the shared GL library of graphics primitives. The widget set used in the original applications was the FORMS library developed by Mark Overmars of the Netherlands for SGI machines using GL graphics. It was anticipated that these widgets would be replaced in the X based Open-GL or Motif versions of the code. However, in February, 1995, T.C. Zhao of University of Wisconsin at Milwaukee announced the release of an X version of the FORMS library which is designated here as XFORMS. It ap-

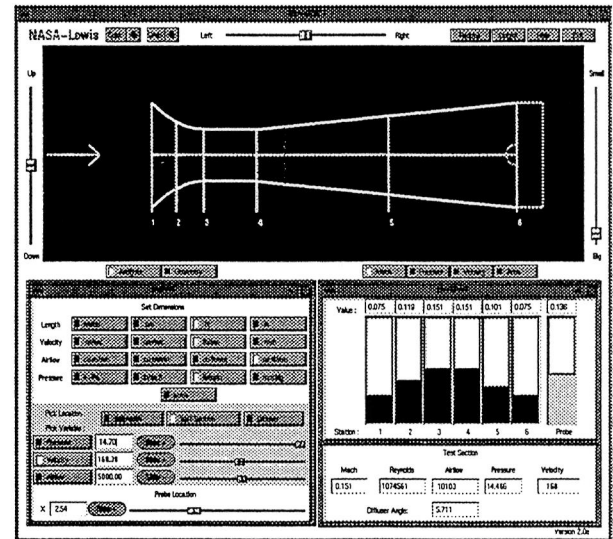


Figure 3: X version of VU-DUCT simulator

peared that the conversion of the applications packages from GL to X graphics would be greatly simplified; the definitions of the widgets would remain essentially the same, some additional coding would be required to account for differences between the two libraries, the analysis routines would remain the same, and only the schematic graphics routines would need to be converted from GL to X. However, even this last modification was easier than expected when it was discovered that the XFORMS library contained many basic graphics functions, like drawing a circle, line or rectangle in X, in an almost identical command format as found in the shared

GL library.

## CONVERSION TO X GRAPHICS

The initial efforts at converting the GL based simulators to X involved simply recreating the old simulators with the new widgets. The results for VU-DUCT are shown in Figs. 3 and 4. The layout of

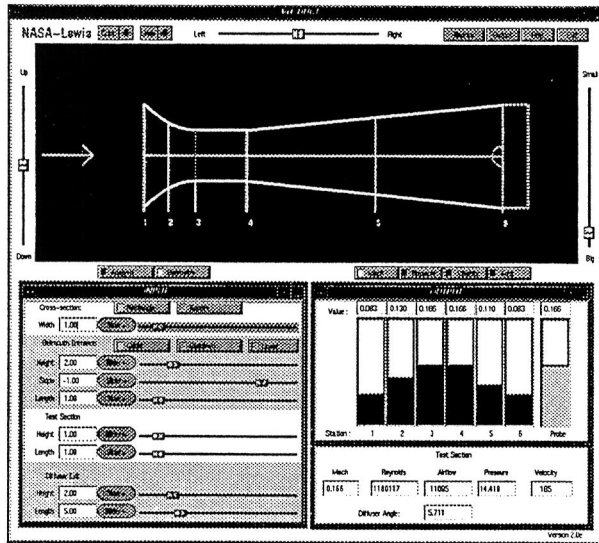


Figure 4: Geometry input panel

this package is identical with that in Fig.1. There are some slight differences in the colors and shapes of the buttons and sliders, but the two packages are nearly identical. Fig. 3 shows the "Analysis" input panel also shown in Fig. 1, while Fig. 4 shows the "Geometry" input panel. The idea behind this grouping was to make all geometry information available to the user on one panel and all other input information available on the other since only one panel could be displayed at a time. In Figs. 3 and 4 information is further grouped by a colored box placed behind the input variables. In Fig. 3, the upper portion of the input panel contains a group of buttons to select the units for display of dimensional data, the middle portion gives flow variable (pressure, velocity) input values, and the lower portion contains widgets to control the flow probe. In Fig. 4, the upper portion of the input panel is used to select cross-sectional shape and width for the entire tunnel while the other three portions control the geometry definition of the bellmouth, test section, and

diffuser respectively. This version was successfully compiled and executed on an SGI, IBM, and Sun workstation in stand-alone mode. However, when the package was executed in a client/server mode between an IBM and SGI workstation a major problem was encountered. The background screen would jump to the foreground whenever the schematic was redrawn. The input and output panels would then be covered by the background panel and could not be accessed.

To solve this problem a major redesign of the graphics portion was required. The background panel was eliminated and replaced by a main control panel as shown in Fig. 5. The schematic view

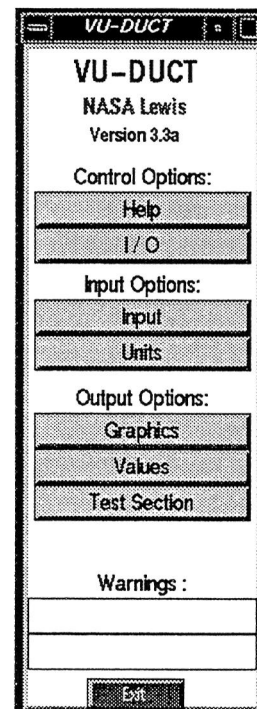


Figure 5: Main control panel

screen of the background panel was changed into a separate panel and is now grouped with the test section and bar graph output panels as output options. The help screen, and recorded I/O panels are now grouped as control options. The two input panels of Figs. 3 and 4 are now regrouped into an input panel and a units panel as input options. All panels are now invoked from the main control panel by clicking on the appropriate buttons. Warning banners which

previously were displayed on the test section panel are also now displayed on the main control panel. Since the background form is no longer used to contain, size and place the other forms, all forms are treated as X windows which can be independently sized, located, displayed, iconified, opened or closed. In some of the applications this presented a problem because the application had originally been developed assuming that only one input panel would be available to the user at a time. Some different logic and re-coding has overcome these problems and allows the user to now freely switch between multiple input panels. In all of the applications the relaxed size constraint permitted more input variables to be added and greatly simplified the layout of the input widgets. Because the display and use of all the panels are now in the control of the user and not the programmer, it is more important to group widgets onto panels according to the function they perform. This move in the direction of more object-oriented programming caused a reorganization of some of the original panels.

The schematic view screen shown at the top of Fig 1. has now become a separate panel with the control sliders attached as shown in Fig. 6. This

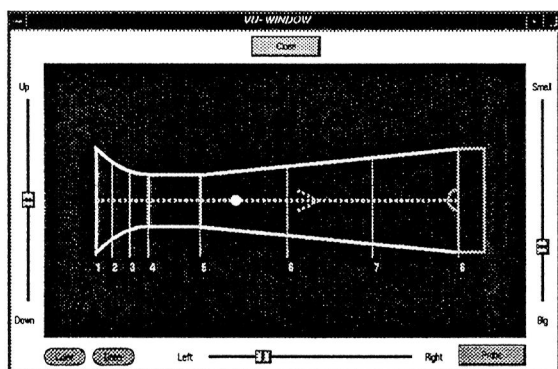


Figure 6: Main viewing form

window is invoked by pushing the "Graphics" button on the main control panel. All of the simulators use this same basic display window, with the graphics picture being different for each problem. For the supersonic simulators, for example, the view window shows shock locations and orientations relative to component surfaces. As the user changes flow conditions the shock angles and intersection locations will change. For the VU-DUCT simulator

shown in Fig. 6, the view window principally shows the shape of the tunnel, the location of the output stations, and the location of the data probe. As the user changes the design of the wind tunnel using the input panel a new geometry is computed and the schematic in the view window is redrawn. The walls of the tunnel appear to move and stretch as the user changes the geometry.

In the original GL version of the code, Fig. 1, the control for the probe location was located on the input panel at the lower left and the flow value at this location was displayed on the output panel at the lower right. In the X version a separate probe panel has been created, as shown in Fig. 7, and all information associated with the probe is controlled and displayed on this panel.. Because the position-

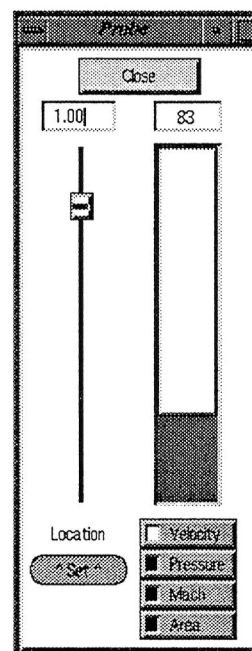


Figure 7: Probe control panel

ing of the data probe requires the operation of the view window, this panel is invoked from the view window using the "Probe" button. The probe location can be set by the input box or slider at the left of Fig. 7 and the location is noted in the view window as a white dot along the dashed centerline. Using the oval button below the locator slider, the user can choose to have the probe move through the tunnel at a fixed speed, or allow the probe to



move at the speed of the local flow. As the probe is moved, the simulator calculates the value of the output flow variable, displays the value on the thermometer gauge to the right, and displays the numerical value above the slider. The user indicates the flow variable choice by the buttons below the output gauge. In the X version, the user may choose to display a different flow variable in the probe panel than that appearing in the output panel; in the GL version the variables must be the same.

The input panel of VU-DUCT shown in Figs. 3 and 4 have been further reorganized in the X version as shown in Figs. 8, 9 and 10. VU-DUCT was orig-

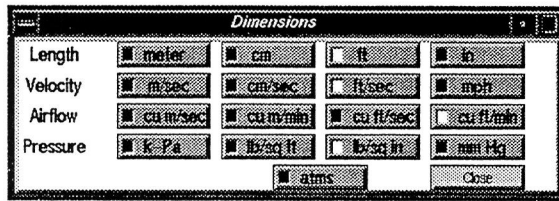


Figure 8: Units panel, Version 1

inally intended for college undergraduates or high school students. Although the analysis is performed non-dimensionally, the results of the analysis are presented to the student dimensionally. The student can pick the dimensions using the units panel shown in Fig. 8. The lighted buttons indicate the chosen combination of dimensions. This button box was originally part of the input panel shown in Fig. 1 and will eventually be copied from the wind tunnel simulator to the basic flow calculators and become a stand-alone program to convert units. The creation of the stand-alone program has been greatly simplified in the X version because of the isolation of the units specification from other input. The lat-

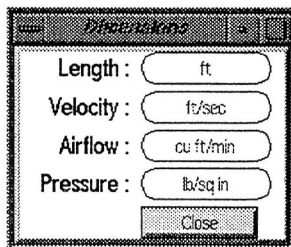


Figure 9: Units panel, Version 2

est version of the code contains an alternate version

of the units panel, as shown in Fig. 9. On this panel, choice buttons are used to select the dimensions and the rows of radio buttons have been eliminated. Clicking the right mouse over the oval choice buttons cause a change in the label and a new selection to be made, clicking the left mouse will produce a menu list of the possible choices. This version of the panel saves space, is easier to modify, and is easier to read and interpret than the previous button box. The only advantage of the button box is that all possible choices remain visible at all times.

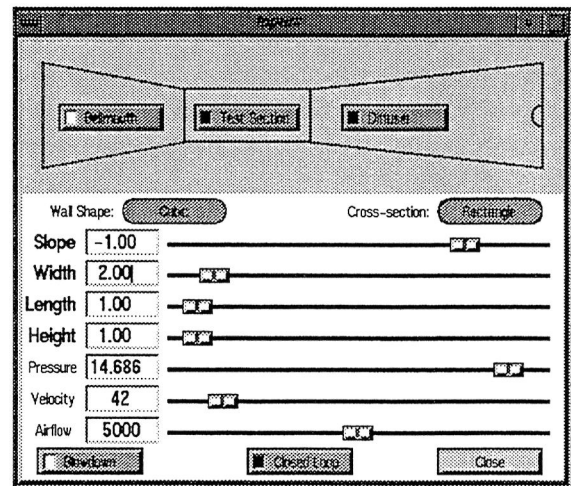


Figure 10: Input panel

The new VU-DUCT input panel, Fig. 10, combines the initial condition specification from Fig. 3 with the geometry specification shown in Fig. 4. In the GL version, input was grouped by flow inputs and geometry with separate panels for each set of input. In the X version, input is grouped by location in the tunnel and only one panel must be displayed. This grouping of information has required some re-coding of the analysis section of the program because more options are now available to the user. The user can now pick any cross section or wall shape and specify any flow value at any location. In the GL version, the user was limited to a single cross section throughout the tunnel. The user now chooses a particular part of the wind tunnel to modify using the buttons in the schematic drawing at the top of the panel, Fig. 10. This schematic does not change with change in geometry, it only serves to indicate which portion of the tunnel is be-



ing modified. Having selected either the bellmouth, test section, or diffuser, the user can change any parameter of the geometry using the sliders, input boxes and buttons in the middle of the panel. The user can also set the value of a flow variable in that section using the sliders or input boxes near the bottom of the panel. The new package can solve for

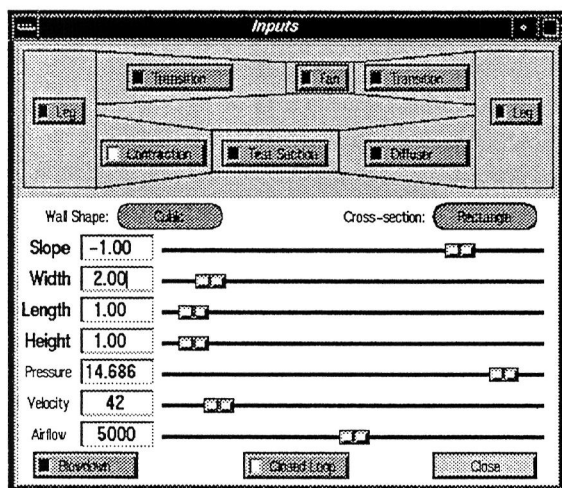


Figure 11: Closed circuit tunnel

flow in an open or closed circuit tunnel with the choice selected by lighted buttons at the bottom of the input form. For the closed circuit tunnel the input schematic changes to show the full circuit as shown in Fig. 11. There are now eight possible sections of the tunnel which can be modified. Within each section, however, the user still has the same choices of geometric and flow variable specification. The grouping by function has greatly simplified the extension of the original package to include closed loop tunnels.

The output panel of Fig. 1 has been split into two panels as shown in Fig. 12. The bar graphs shown in the top of Fig. 12 move as the calculated conditions in the tunnel change with either change in geometry or input flow conditions. The panel at the bottom of Fig. 12 gives numerical conditions in the test section. Variations of the output panel are used in all the simulators since the moving bar graphs show the designer both the amount of change (large or small) of a variable and the direction in which that change occurs (increase or decrease). The numerical values are also available on the form above

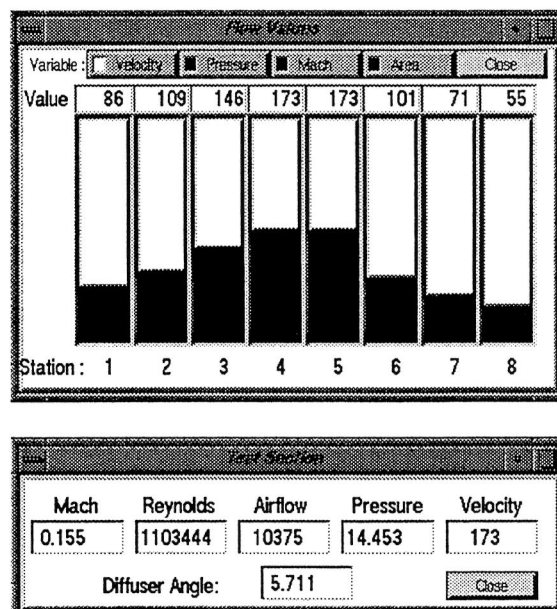


Figure 12: Output panels

the bar graphs. In the panel shown in Fig. 12, the value of a single variable at multiple locations is displayed. The user chooses the variable to display by the buttons at the top of the form and the location is noted on the schematic drawing of the view window, Fig. 6. In the inlet simulator the output of many flow variables at a single location are also displayed on a separate panel. With the X version of the simulator, the designer can invoke both types of output and place them on the workstation screen simultaneously which was not available with the GL version.

Having considered several of the panels individually, Fig. 13 shows how these types of panels might appear together on the computer screen. This figure is a screen dump from a session with the new DUCTSIZER simulator. DUCTSIZER determines the size and pressure losses through a series of connected ducts given the flowrate and the length of the various sections. DUCTSIZER uses the equal friction method to perform the analysis. The upper portion of Fig. 13 shows the layout of nine duct sections from a plenum at the left. The user can specify the equivalent length and flow rate through each section using the two input panels at the lower left. The code will then determine the pressure loss through the system, the velocity in each section and

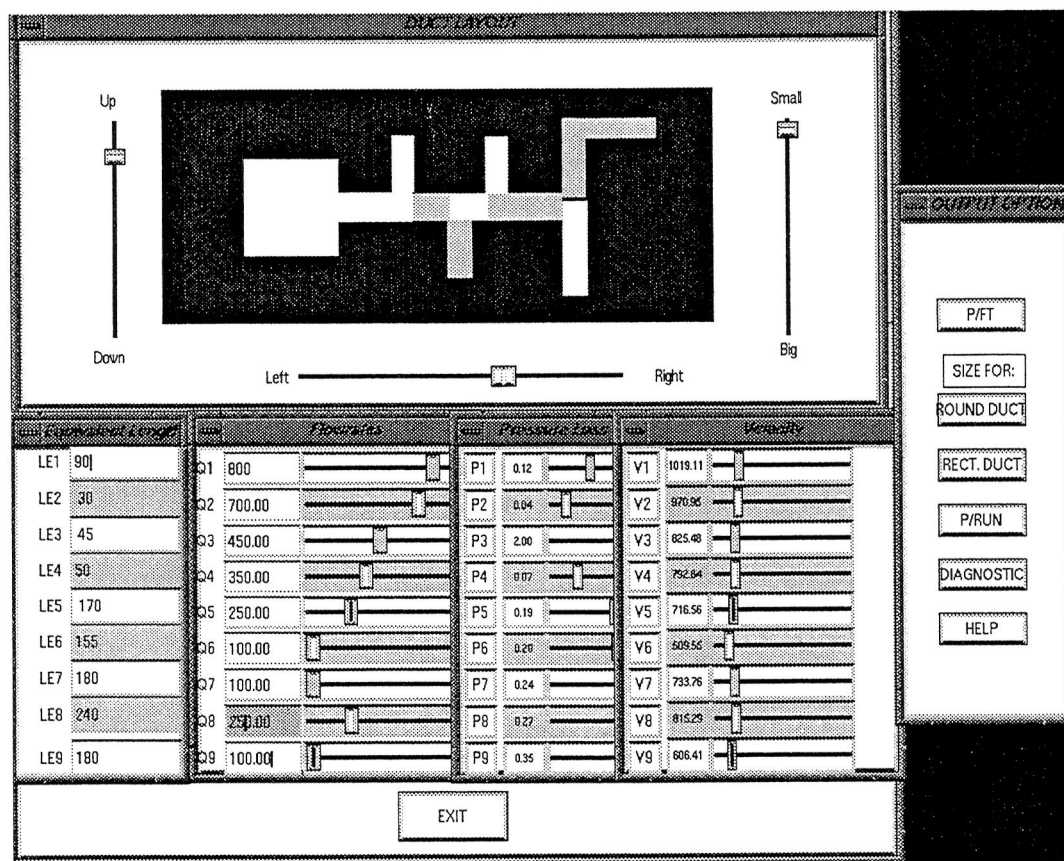


Figure 13: Screen dump from DUCTSIZER

the size of each duct and display this information in the two output panels at the lower right and in the schematic drawing at the top. In the actual display, the duct section in the schematic drawing and the appropriate input and output sliders are color coded for quick identification. In this figure they are only shown in black and white. The user can invoke additional output panels, diagnostic and help screens by using the buttons on the control panel at the right. The arrangement of the various windows on the screen is at the user's discretion.

## SUMMARY

Six computer applications have been developed to solve various aerodynamic problems using interactive computer graphics. All six applications have recently been converted from a GL based graphical interface on an SGI machine to an X based graphical interface that runs on a variety of workstations.

The X based programs can also be implemented in the client/server mode across multiple platforms. The change in graphics mode has brought with it a change in the layout, logic, and options available in all the applications. This paper has concentrated on the evolution of one of the applications, VU-DUCT, and has shown a more logical grouping of information based on function. The resulting package has been easier to modify and extend to include additional problems than the previous GL version.

## ACKNOWLEDGEMENTS

The current simulator uses the XFORMS library of GUI developed by Mark H. Overmars, Department of Computer Science, Utrecht University, The Netherlands and T.C. Zhao of the University of Wisconsin at Milwaukee. The XFORMS library is copyrighted by T.C. Zhao and Mark Overmars but can be used for non-commercial and non-profit purposes.

The software developed by the author is in the public domain and may be copied and also used for non-commercial products, but not resold. Copies of the source are available from the author at the NASA Lewis Research Center.

## REFERENCES

1. Koenig, K. and Hodge, B.K., "A Suite of Personal Computer Programs to Support Propulsion Education", AIAA 93-2053, June, 1993.
2. Hodge, B.K. and Koenig, K., "SOLROC, A Solid Rocket Motor Internal Ballistics Software Element", AIAA 94-3115, June, 1994.
3. Torella, G. and Lombardo, G., "Computer Codes for the Training on Auxiliary Power Units (A.P.U.)", AIAA 94-3113, June, 1994.
4. Benson, T.J., "An Interactive Educational Tool for Compressible Aerodynamics" AIAA 94-3117, June, 1994.
5. Benson, T.J., "An Interactive, Design and Educational Tool for Supersonic External Compression Inlets", AIAA 94-2707, June, 1994.
6. Benson, T.J., "A Workstation Based Simulator for Teaching Compressible Aerodynamics", AIAA 95-0070, January, 1995.
7. Benson, T.J., "Interactive Computer Graphics Applications for Compressible Aerodynamics", AIAA 95-0119, January, 1995.
8. Benson, T.J., "An Interactive Educational Tool for Turbojet Engines", AIAA 95-3055, July, 1995.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1995		3. REPORT TYPE AND DATES COVERED Technical Memorandum
4. TITLE AND SUBTITLE X Based Interactive Computer Graphics Applications for Aerodynamic Design and Education			5. FUNDING NUMBERS  WU-505-62-52	
6. AUTHOR(S) Thomas J. Benson and C. Fred Higgs III				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191			8. PERFORMING ORGANIZATION REPORT NUMBER  E-10052	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Washington, D.C. 20546-0001			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  NASA TM-107128 AIAA-96-0049	
11. SUPPLEMENTARY NOTES Prepared for the 34th Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 15-18, 1996. Thomas J. Benson, NASA Lewis Research Center, and C. Fred Higgs III, Rensselaer Polytechnic Institute, Troy, New York. Responsible person, Thomas J. Benson, organization code 2670, (216) 433-5920.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Unclassified - Unlimited Subject Category 64  This publication is available from the NASA Center for Aerospace Information, (301) 621-0390.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  Six computer applications packages have been developed to solve a variety of aerodynamic problems in an interactive environment on a single workstation. The packages perform classical one dimensional analyses under the control of a graphical user interface and can be used for preliminary design or educational purposes. The programs were originally developed on a Silicon Graphics workstation and used the GL version of the FORMS library as the graphical user interface. These programs have recently been converted to the XFORMS library of X based graphics widgets and have been tested on SGI, IBM, Sun, HP, and PC-Linux computers. The paper will show results from the new packages, detailing the change in design and philosophy brought about by the move to X based graphics using the new VU-DUCT program as a prime example. VU-DUCT has been developed as an educational package for the study of subsonic open and closed loop wind tunnels.				
14. SUBJECT TERMS Computer graphics; Education; Aerodynamics			15. NUMBER OF PAGES 12	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	



National Aeronautics and  
Space Administration  
**Lewis Research Center**  
21000 Brookpark Rd.  
Cleveland, OH 44135-3191

Official Business  
Penalty for Private Use \$300

POSTMASTER: If Undeliverable — Do Not Return

