AEDC-TR-90-25

C#3

Interactive EAGLE: An Interactive Surface Mesh and Three-Dimensional Grid Generation System

Version 1.0, User's Guide

W. E. Dietz
and
S. B. Evans
Calspan Corporation/AEDC Operations

December 1990

Final Report for Period October 1, 1988 through June 30, 1990

Approved for public release; distribution is unlimited.

**ARNOLD ENGINEERING DEVELOPMENT CENTER
ARNOLD AIR FORCE BASE, TENNESSEE
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE**

NOTICES

APPROVAL STATEMENT

This report has been reviewed and approved

Mark S. Briski

MARK S. BRISKI, Capt, USAF
Directorate of Technology
Deputy for Operations

Approved for publication:

FOR THE COMMANDER

Keith L. Kushman

KEITH L. KUSHMAN
Technical Director
Directorate of Technology
Deputy for Operations

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1 AGENCY USE ONLY (Leave blank)

2 REPORT DATE
December 1990

3 REPORT TYPE AND DATES COVERED
Final, October 1, 1988 -- June 30, 1990

4. TITLE AND SUBTITLE
Interactive EAGLE: An Interactive Surface Mesh and Three-Dimensional Grid Generation System

5 FUNDING NUMBERS
PE65807F

6 AUTHOR(S)
Dietz, W. E. and Evans, S. B., Calspan Corporation/AEDC Operations

7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Arnold Engineering Development Center/DOT
Air Force Systems Command
Arnold AFB, TN 37389-5000

8 PERFORMING ORGANIZATION REPORT NUMBER
AEDC-TR-90-25

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Arnold Engineering Development Center/DO
Air Force Systems Command
Arnold AFB, TN 37389-5000

10 SPONSORING/MONITORING AGENCY REPORT NUMBER

11 SUPPLEMENTARY NOTES
Available in Defense Technical Information Center (DTIC).

12a DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b DISTRIBUTION CODE

13 ABSTRACT (Maximum 200 words)
The generation of surfaces and three-dimensional grids for use in finite-difference computations is tedious and time-consuming, but can be made more efficient when implemented in a workstation-based interactive environment. Interactive EAGLE is an interactive surface mesh and three-dimensional grid generation system based on the well-known EAGLE grid generation codes. Interactive EAGLE implements most of the functions available in the batch EAGLE codes and has been extended to allow input in the form of cross-sectional profiles. Interactive EAGLE performs all input functions through interactive screens and menus, and significantly increases the rate at which surface meshes and three-dimensional grids can be generated. In addition, the user can access the wide range of powerful operations available in the EAGLE codes without knowledge of the input requirements and internal structure of the batch versions.

14. SUBJECT TERMS
grid generation
computational fluid dynamics
finite-difference techniques
surface generation
EAGLE

15 NUMBER OF PAGES
171

16 PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18 SECURITY CLASSIFICATION OF THIS PAGE	19 SECURITY CLASSIFICATION OF ABSTRACT	20 LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	**SAME AS REPORT**

COMPUTER GENERATED

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102

PREFACE

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC). The work was performed under AEDC Project Number DC71, at the request of the AEDC/DOT. The AEDC Project Manager was Capt. Mark Briski. The results were obtained by Calspan Corporation, AEDC Operations, operating contractor for the aerospace flight dynamics testing facilities at the AEDC, AFSC, Arnold Air Force Base, Tennessee. The work was conducted during the period October 1, 1988 through June 30, 1990. The manuscript was submitted for publication Nov. 8, 1990.

The authors acknowledge the contributions of Dr. John C. Erickson, Dr. Brian R. Circelli, and Mark Rist for their editorial and technical assistance during the preparation of this report.

CONTENTS

ILLUSTRATIONS

Figure | Page

1.0 INTRODUCTION

The generation of three-dimensional grids for use in fluid dynamics computations is a tedious and time-consuming process, primarily because it is inherently iterative in nature. Surface meshes and three-dimensional grids often undergo many refinements before they are suitable for use in a computation. In conventional batch operations, each refinement requires a complete execution of a surface or mesh generating program, only after which the surfaces and meshes can be examined for correctness. However, in an interactive environment the user can continually examine and correct the surfaces and meshes as they are developed. Therefore, the surface and mesh generation process can be made much more efficient when implemented in a workstation-based interactive environment.

Interactive EAGLE is based on the EAGLE surface mesh and three-dimensional grid generation programs developed by J. F. Thompson et al. (Refs. 1 through 11). Interactive EAGLE implements most of the functions available in the batch EAGLE codes and has been extended to allow input in the form of cross-sectional profiles. Interactive EAGLE performs all input functions through interactive screens and menus. In the interactive environment, the user becomes an integral part of the iterative grid generation process, thereby greatly increasing the rate at which surface meshes and three-dimensional grids can be generated. In addition, the user can access the wide range of powerful operations available in the EAGLE codes without knowledge of the input requirements and internal structure of the batch versions.

An important feature of Interactive EAGLE concerns the preservation of the process (i.e., the sequence of graphical construction operations that are invoked by the user) whereby surfaces and grids are generated. The practical result of this feature is that no interactive session ever needs to be repeated in its entirety in order for changes to be made to a previously generated system of surfaces or grids. This has been found to be a major factor in increasing the throughput of the surface and grid generation process.

The Interactive EAGLE system consists of two separate codes. The first generates the boundary surface configuration (Ref. 3); the second generates a grid within the field (Ref. 4). The entire boundary of the physical region is constructed incrementally by the boundary code, which creates a file of surfaces for input to the grid code. The grid in the field is then generated by the grid code. The generated grid can then be viewed with a grid viewing system such as PLOT3D (Ref. 12).

This report documents, and is restricted to, the operation of the Interactive EAGLE surface and grid generation systems. Readers who are interested in various aspects of surface and grid generation technology are referred to Refs. 1 through 11. In particular, those who are interested in the technical aspects and operation of the batch EAGLE codes are directed to Refs. 2, 3, and 4.

This document is organized in the following manner: Sections 2.0 through 2.4 introduce the general concepts applied to the Interactive EAGLE surface and grid generation codes and describe the methods by which the graphical interfaces were implemented into the batch EAGLE codes. The "user's manual" portion of this report includes Sections 3.0 and 4.0, which describe the capabilities and interactive operation of the surface and grid generation functions, respectively. Sections 3.0 through 3.5 describe the general operation of the surface generation system, and provide details about the mechanisms of selecting options, manipulating the graphical images, and producing hard-copy plots. The functions that may be used to generate primitive graphical constructs such as points, curves, and surfaces are described in detail in Section 3.6. The operation of the three-dimensional volume grid generation system is described in Section 4.0. The final section, Section 5.0, provides an example of the application of the Interactive EAGLE surface and grid generation codes to an aircraft fuselage configuration.

2.0 GENERAL CONCEPTS

Interactive EAGLE incorporates several innovative concepts that greatly enhance the surface mesh and grid generation process. The major feature is the system of interactive graphical interfaces through which the user invokes graphical transformation functions and manipulates graphical images. Another major feature is the method of implementing the graphical interfaces, which allows new versions of EAGLE to be incorporated into the interactive system with few modifications. In addition, the implementation method preserves the process whereby the user generates the desired product; this feature has been found to be a major factor in increasing the efficiency of surface and grid generation.

2.1 IMPLEMENTATION OF THE INTERACTIVE INTERFACES

The interactive interfaces were implemented using Fortran 77 versions of the batch EAGLE surface and volume grid generators. Initial interface implementation was performed on an Apollo DN580 running AEGIS. The graphical interfaces were written using the Apollo GMR3D graphics library, primarily to obtain maximum benefit from the hardware, but also because of the close resemblance of GMR3D to PHIGS (Programmers Hierarchical Interactive Graphics System). PHIGS, as of this date, appears to be an emerging standard. PHIGS versions of the surface and grid generators have been developed.

The interfaces were implemented in a manner that minimizes changes to the original EAGLE codes. This was readily accomplished because of the structure of the batch EAGLE codes. The EAGLE codes are, in effect, libraries of routines that transform one graphical construct into another, usually more complex, graphical construct. EAGLE executes particular routines in response to a user-defined command file, which is a list of commands and associated user data that perform a sequence of desired graphical transformations. The input and output streams of the batch and interactive versions of EAGLE are depicted in Figs. 1a and b, respectively.

The batch version of an EAGLE code (either the surface generator or the grid generator) interacts only with its command file; the command file is generally written using an ordinary word processor. The data flow is one way, from the command file to the EAGLE code. In the interactive version, data flow forms a loop between the EAGLE code, a library of interactive interfaces, and the command file. During a step in an interactive session, EAGLE will first invoke the graphical interfaces. Through user input, a line (or possibly several lines) of the command file is built by functions in the interfaces. Once a line is added to the command file, it is processed by what is essentially a batch version of the EAGLE code. Once processing is completed, control returns to the graphical interfaces. In this manner, the user is made an integral part of the iterative surface and grid generation process.

The required modifications to the batch EAGLE codes consist of relatively few lines of code, which simply call the interfaces at an appropriate location within the code. This design approach allows new versions of the batch EAGLE codes to be easily incorporated into the interactive environment. In addition, since the original input format of the batch codes was left intact, batch EAGLE input is compatible with the interactive EAGLE codes.

2.2 FORMS AND MENUS

The Interactive EAGLE system functions through a sequence of construction operations, each of which corresponds to one or more of the input operations in the original batch versions of the EAGLE codes. Each of these construction operations is represented symbolically on a graphics screen through which the various quantities involved in the operation are entered. For the purposes of maintaining a consistent nomenclature, the interactive interfaces that invoke functions will be referred to as "forms."

There is a form for every major line, surface, and volume grid generation function in the EAGLE codes. A particular form is invoked by selecting menu options. After the form appears, the user fills in all variables and options that define the particular geometric element the user wishes to create. For instance, the form that accepts data for generating cubic curves (See Section 3.6.2.2) contains data slots for defining endpoints, slopes, and number of points along the curve. After the user fills out the screen options, a box on the screen is selected that informs the system that input has been completed. The interface function then generates the appropriate batch EAGLE commands corresponding to the input provided by the user. Control is returned to the batch EAGLE code, which then processes the commands generated by the interactive interface. After processing, the new geometric element is displayed in a graphics window in the lower left corner of the screen. Control is returned to the particular form that generated the last processed command. The user can then traverse the menu hierarchy to select and execute another function.

Depending on the user input, a form may generate several lines in the command file. New graphical information is not displayed until all lines of input associated with a form are processed by EAGLE. In effect, (i.e. to the user), a single form invokes a single function, even though internally several EAGLE functions may have been performed.

2.3 PROCESS PRESERVATION

Many mesh generation systems regard the final surface mesh or grid as the end product. However, the process (i.e., the sequence of functions invoked and associated data provided by the user whereby a surface or grid is generated) is of equal importance. The Interactive EAGLE system naturally preserves all steps of the generation process when the command file is built via the forms. The preservation of the process can result in large savings in time and effort when alterations must be made in attributes of surfaces and grids. For example, geometric attributes such as the location of points, the number of points on a curve or surface, and spacing, etc., can be predefined at a single location in an interactive session and subsequently associated with various surfaces and grids. As a result, any changes in geometric attributes can be made in one location in the preserved process; if the process is then re-executed, the changes will be automatically propagated through all surfaces and grids that are associated with those attributes. The preservation of the surface and grid generation process also allows the generation process to be created on a workstation and the resulting run stream to be executed on another computer in batch mode, instead of on the workstation, if desired (See Fig. 1).

The command file generated by the interfaces is referred to as a "restart" file. The restart file can be accessed at a later time and processed, either all at once or one step at a time. In the stepping mode, a user can view incremental changes and additions to the geometric elements and identify steps that require modification. The restart file can then be modified interactively and reprocessed. In addition, lines in the restart file may be skipped if required. Because the surface and volume grid generation process is saved, no interactive session ever needs to be repeated entirely.

Input errors are detected when made and thus can be corrected immediately. Since the sequence of input operations that produce a geometric construct is only written to the restart file after successful execution of the construction operation, errors may be corrected without the erroneous input being preserved. It is also possible to switch the restart file writing operation to a manual mode whereby the sequence is written to the restart file only after the construction is accepted by the user. This allows the results of each construction operation to be examined before being made a part of the preserved generation process.

2.4 SEGMENT AND REFERENCE NUMBERS

Geometric constructs, such as points, curves, and surfaces, are each assigned a unique label, which in the EAGLE codes is a positive integer. Geometric attributes such as spacings and numbers of points on curves may also optionally be identified with a label. Geometric constructs may be identified either directly by supplying the label from the keyboard, or can be identified graphically. Graphical identification eliminates the necessity of having to remember large numbers of labels. Therefore, keyboard entries are required only once when one of those elements is defined, since identification can be made graphically in all later usage of the element. Keyboard input is thus kept to a minimum.

When initiating an interactive session, an experienced user will generally develop tables of geometric objects and attributes that can be referenced later. All table entries can be referenced later in the interactive session by their positions (i.e. their indices) in the tables. The table indices thereby serve as labels or identifiers of the related geometric objects and attributes. The advantage of indirect referencing is evident when changes must be made in a completed surface or grid. A user can, for instance, change the dimensions of a group of similarly dimensioned surfaces by simply changing the appropriate table entries. When the restart file from the session that originally generated the surfaces is executed again, the surfaces will be automatically regenerated with the new dimensions.

3.0 SURFACE GENERATOR — GENERAL OPERATION

3.1 START-UP PROCEDURE

To execute the surface program, a user will enter the subdirectory in which all surface and grid geometries are to be stored. When the surface generation program is executed on an Apollo, the user will be prompted to request either direct or borrow mode. Direct mode opens a window in which the surface generator operates; the user can open other windows during the operation of the surface generator. In borrow mode, the program will take up the entire screen, and all windowing operations will be inhibited until execution of the surface generator is terminated.

3.1.1 Selection of Menu and Form Options

Menu options are selected by first placing the cursor on the appropriate menu option. The option box will be highlighted, and any text in the box will be italicized. Selection of an option is performed by pressing the left mouse button.

Form data is entered via the mouse and keyboard. When the cursor is placed on a form data box, the box will be highlighted in green, indicating it is ready for data to be entered. The data may then be entered through the keyboard or, in some cases, by simply pressing the RETURN key (See Section 3.3). Toggle boxes in forms are selected by simply passing the cursor over the box. The toggle box will change color to indicate that a selection has been made.

3.2 GLOBAL FORM MENU OPTIONS

The following options appear on virtually every form in the menu hierarchy:

First Menu Option

The first menu option of each form is a box identified with the name of the form. For example, the form that generates straight lines has a menu option called Line (See Section 3.6.2.1). If this option is selected, the entire form is reset to its initial condition. All default values are reset, and the form is ready for processing. This option should be selected if an error is made while entering data. It should also be invoked if the form is to be used more than once without exiting. It is not necessary to invoke this menu option when a form is selected from a menu higher in the hierarchy.

Clear Side Plot

Selection of this option clears the graphics window. After the Clear Side Plot option is selected, only newly generated segments will be displayed unless the user specifically requests the display of previously generated segments.

Restart Save/No Save

When a form is filled out and executed, the actions are normally saved in the restart file. A user may wish to disable this action, in which case this option may be selected. The current state of this option will be shown in a small box that occurs in the lower right corner of each form, which will display either WRITE TO RESTART FILE, or OMIT FROM RESTART FILE. If an error occurs during the processing of a restart file or form, the subsequent writing to the restart file will be automatically disabled until the user invokes this option.

Help Window/Run Stream

A small help window is ordinarily displayed during an interactive session. The help window guides the user in entering data into the form. Advanced users may wish to see the commands generated by the execution of a form. This option allows a user to display either the help window or the EAGLE command file as it is generated.

Last Menu Option

The last option on each form will return control to one level up in the menu hierarchy. On most forms, this option is simply EXIT. Depending on the location within the hierarchy, the option may read EXIT PROGRAM (top level), RETURN TO MASTER (second level), or RETURN TO PRIMARY (profile functions).

3.3 COMMON FORM INPUT BOXES

Each form has several input boxes and buttons that are common to virtually all forms. They are defined as follows for conciseness:

Segment

This input box contains the segment number of the geometric construct to be generated by the form. For instance, a surface can be generated by blending two curves. The segment number of the surface that is to be generated should be input into this box. The output segment can be named automatically by simply placing the cursor in the SEGMENT box and hitting the RETURN key. The next available segment identifier will appear in the SEGMENT box and will be associated with the output geometry.

File

The function associated with a form will store the generated segment in memory, if an identifying integer has been entered in the SEGMENT input box (See previous paragraph). However, the segment may be written to a file by entering an integer in the FILE input box. If a positive integer is placed in this box, a file named FTxx will be written, where xx is the input integer plus ten. The file may be subsequently read by the Read Segment-File option described in Section 3.6.4.7, or by the interactive grid generator described in Section 4.0. Any previously written file by that name will be overwritten. Note: The FILE and SEGMENT input boxes cannot be used simultaneously.

Spacing

Some forms that allow spacing along a curve or surface to be specified have input boxes similar to that depicted in Fig. 2. These boxes allow the user to specify the spacings at the endpoints of curves. There are different methods in which the spacing information may be entered as follows:

Spacing or Number — This allows the user to specify the actual spacing value, or a pointer referencing a spacing table entry (See Section 3.6.1.3). Actual spacing values are entered directly into the SPACING box. Table entry values may be examined by placing the cursor in the NUMBER box. Pressing the J and K keys will decrement and increment, respectively, the spacing indices that have been previously defined; the corresponding spacing values will appear in the SPACING box.

Other Segment — The user may enter the number of a previously defined curve segment and specify the end from which the spacing is to be obtained. The segments may be reviewed by placing the cursor in the OTHER SEGMENT box. Pressing the J and K keys will decrement and increment, respectively, the segment numbers that have been defined. The segment corresponding to the number in the box will be highlighted in the graphics area. The desired segment may be selected by pressing the RETURN key. The appropriate end of the segment is selected by placing the cursor over the appropriate box (i.e., either the FIRST or LAST box) and pressing the left mouse button.

Relative or Absolute — These boxes are toggles that select whether relative or absolute spacing is to be imposed. Relative spacing assumes the arc length of a curve is unity; a relative spacing value of 0.0015, for instance, would place a point 0.15 percent of the total arc length from an endpoint. An absolute spacing value of 0.0015 would place a point 0.0015 physical units from an endpoint.

If no spacing information is provided by the user, points will be spaced evenly along the length of a curve, or equiangularly in the case of curves obtained from conic sections.

Points or Number

Many forms have a box that looks like the diagram depicted in Fig. 3. The POINTS box refers to the actual number of points the user wishes to specify along a line or curve. The NUMBER box will accept as input an index into a previously defined points table (See Section 3.6.1.2). Either method may be used, but the latter method is more convenient if subsequent changes are to be made to the restart file. The numbers of points that have been previously defined in a table may be reviewed by placing the cursor in the NUMBER box. Pressing the J and K keys will decrement and increment, respectively, the indices referencing numbers of points. The number of points corresponding to the index will appear in the POINTS box.

Segment and Indices

Some forms have boxes that look like the diagram depicted in Fig. 4. This form entry box allows a point to be specified on an existing curve or surface segment by specifying the segment number and the indices on the segment. When data are entered in these boxes, the referenced segment will be highlighted in the graphics area, and the specified point will be identified with a circle. Segment numbers may be decremented or incremented automatically by pressing the J or K keys, respectively, when the cursor is in the SEGMENT box; the corresponding segment will be highlighted in the graphics area. The left and right INDICES boxes define the indices on a curve or surface; by convention, the left and right indices are referred to as "faster-running" and "slower-running" indices, respectively. The faster-running index may also be decremented and incremented with the J and K keys; the slower-running index may be incremented with the I key and decremented with the M key. As the indices are incremented and decremented, the corresponding point on the segment will be highlighted with a white circle.

Do It

The DO IT button appears on almost every form, and is selected when all input to a particular form has been completed. The DO IT button is selected by placing the cursor on top of the button and pressing the left mouse button. When the information on the form is processed, all of the corresponding input statements of the EAGLE batch run stream that are involved are written into the restart file to preserve the generation process. The input statements are executed by EAGLE, and any visible changes to the current geometric construct are updated in the graphics window.

3.4 CONTROL OF THE GRAPHICS DISPLAY

The capital B and R keys may be used at any time to toggle the size of the graphics area, or to center the graphical image in the graphics area, respectively.

B — This key toggles the size of the graphics area. The graphics area is either in the lower left of the screen, or it utilizes the entire screen.

R — This key scales and centers the graphical image to fit within the graphics display area, at its original position. This option only shrinks the graphics information; it will not enlarge it to utilize the entire display area.

The image in the graphics area can be manipulated by appropriate movement of the mouse. The cursor must be in the graphics area for these operations to be active.

Pan — Press the right mouse button and move the mouse.

Zoom — Press the middle mouse button and move the mouse up or down.

Rotate — Press the left mouse button and move the mouse, or press the middle mouse button and move the mouse left or right.

The sensitivity of the mouse movement may be adjusted to any desired level while the cursor is in the graphics area. Pressing the I key increases the mouse sensitivity, and pressing the M key decreases sensitivity. The keys may be pressed as many times as is necessary to achieve the desired level of control.

3.4.1 Postscript Plots

A Postscript plotting file may be generated at any time during the operation of the surface generator. Placing the cursor in the graphics window and pressing the P key will generate a file containing only the graphical image; pressing the key while the cursor is in a menu box will generate a file containing the entire screen. The Postscript files are named POSTxxx.DAT, where xxx begins with 001 and is incremented each time the P key is pressed. The files can then be processed by a Postscript-compatible printer.

3.5 MASTER MENU

The master menu, depicted in Fig. 5, appears when the surface generator is invoked. The options that are selectable from the master menu are discussed in the following sections.

3.5.1 Restart File

The menu items depicted in Fig. 6 appear when the Restart File option is selected. The Restart File operations allow the user to read and process a file that contains a list of EAGLE commands. This file most likely would have been created during a prior interactive session, but could also be a batch EAGLE command file. As the commands are processed, the graphics area is automatically updated. The user is prompted for an output file in which to store the processed commands. The options available under this menu item are as follows:

Restart File Name

This option allows the user to specify the name of the restart file. When this option is invoked, the system prompts the user to enter the name of a file consisting of EAGLE commands. The file name must follow the file naming conventions for Fortran 77. The file must also be located in the currently assigned directory. When the file name is entered, the command file appears on the screen, as illustrated in Fig. 6.

Process Entire File

This option allows the user to process the entire input file sequentially. The graphics area in the lower left corner of the screen will be updated as each command line is processed.

Next Instruction from File

Selection of this option allows the user to process only the next instruction in the input file. The instruction between the two horizontal lines depicted in Fig. 6 will be performed if this option is selected. This "stepping" mode is used if a user intends to modify a subsequent command line.

Skip Instruction from File

This option allows the user to skip over the next instruction in the input file, in the event the user wishes to modify the instruction through a form. The modified instruction is written in place of the skipped instruction.

Next Instruction from Input

This option allows the user to manually type in an EAGLE surface command line. Proper use of this option requires knowledge of batch EAGLE input, and therefore should be avoided by novices. For valid commands, refer to Refs. 2, 3, and 4.

3.5.2 Screens

All of the geometric functions are located in the SCREENS menu hierarchy. Selection of the SCREENS option allows the user to access and execute all EAGLE curve and surface generation functions in addition to various utility functions and the profile processing enhancements. The SCREENS menu hierarchy is described in more detail in Section 3.6.

3.5.3 Plot Segments

This option provides the capability to display predefined points or segments in a specified color. The screen shown in Fig. 7 is displayed when the Plot Segments option is selected. The following information must be provided:

Mode Selection — The user must choose between displaying predefined points or segments. The user makes this choice by moving the cursor over the desired toggle box; this action will highlight the toggle box.

Range — The range is specified by entering positive nonzero integer numbers defining the range of points or segments the user wishes to display.

Color Selection — The user may choose the color in which to display the selected information by simply moving the cursor on top of the desired color box. The last color box touched will be the designated color. The segment generated last is highlighted.

3.5.4 Clear Plot

This option allows the user to remove the graphics display area from the screen. Only newly generated segments are displayed after the Clear Plot option is selected, unless the user specifically requests the display of previously generated segments.

3.5.5 Set Screen Colors

This option allows the default screen color attributes to be changed. The screen shown in Fig. 8 is displayed when the Set Screen Colors option is selected. The following information must be provided:

Attribute Selection — The user can choose between five different screen attributes,

Help Window Color

Prompt Highlight Color

Toggle Highlight Color
Help Window Text Color
Form Labels Color

Color Selection — The user may choose the color in which to display the selected information by moving the cursor on top of the desired color box. The last box touched will be the color in which the chosen attribute will be displayed. The new colors are applied after this option is exited.

3.6 SURFACE GENERATION FUNCTIONS

The surface generation functions are accessed by selecting the Screens option from the master menu. The menu depicted in Fig. 9 is displayed when the Screens option is invoked. The various options are discussed in the following sections.

3.6.1 Settings

This option allows (1) point coordinates, (2) the number of points along a curve, and (3) spacings to be defined for subsequent indirect referencing. Almost all forms that construct geometric objects allow these values to be referenced by their positions in the tables (i.e., by their indices, if the tables are treated as one-dimensional arrays). The menu depicted in Fig. 10 is displayed when this option is selected.

3.6.1.1 Set Point

This option allows points to be defined, either by directly entering the Cartesian coordinates of each point, or by referencing a point on an existing segment. Each defined point is associated with an index defined in the POINT column. The screen depicted in Fig. 11 is displayed when this option is selected.

The indices in the POINT column may be automatically incremented if the RETURN key is pressed when the cursor is in the POINT column. Points are entered either by specifying coordinates in the X, Y, and Z columns or by entering segment numbers and point indices. The SEGMENT and INDICES boxes have special features (See Section 3.3). Points are plotted in the graphics window as they are defined. In the example depicted in Fig. 11, three points have been entered into the table. These points will be referenced later in subsequent examples.

3.6.1.2 Points

This option allows a table to be built that associates the number of points along a curve or surface segment with an index. These indices, defined in the NUMBER column, can be referenced later in an interactive session. The screen depicted in Fig. 12 is displayed when this option is selected.

The indices in the NUMBER column may be automatically incremented if the RETURN key is pressed when the cursor is in the NUMBER column. In the example depicted in Fig. 12, four numbers have been entered in the POINTS column. The values entered in this example will be referenced in subsequent examples. Since no geometric segments have been defined, the graphics area is not affected by the building of the table.

3.6.1.3 Spacing

This option allows a table to be built that associates spacings on a curve or surface with an index. These indices, defined in the NUMBER column, can be referenced later in an interactive session. The screen depicted in Fig. 13 is displayed when this option is selected.

The NUMBER column may be automatically incremented by pressing the RETURN key while the cursor is in the NUMBER column. Four spacings have been defined in the example depicted in Fig. 13. These values will be referenced in subsequent examples. The graphics area is not affected, since no segments are defined by the table entries.

3.6.1.4 Composite Points

The Composite Points option is a more sophisticated version of the Points option described in Section 3.6.1.2. The screen depicted in Fig. 14 is displayed when the Composite Points option is invoked.

Curve segments may be defined by concatenating smaller segments or by extracting segments from larger segments. This option performs the arithmetic necessary to calculate the number of points on segments that are generated through concatenation or extraction functions. There are five toggles that can be selected while building this table. They are defined as follows:

SUM − 1 The sum of all points in the curves
− the number of curves
+ 1

DIF + 1 The sum of all points in the curves
− the number of points in each extracted segment
+ the number of extracted segments

SUM The sum of all points in the curves

DIF The number of points in the first curve
– the sum of the number of points in the remaining curves

PROD The product of points in each curve

As an example, consider a curve segment that is built by the concatenation of three curve segments containing 4, 6, and 5 points, respectively. The sum of all points in the curves is 15, and the number of curves is 3. The number of points in the final segment is 13, since the endpoints of the middle segment are coincident with endpoints of the other segments. This value is calculated by selecting the SUM – 1 toggle. Other values may be calculated from the same input by the selection of other toggles.

In the example illustrated in Fig. 14, a value associated with index 5 is generated by (1) selecting the SUM toggle, (2) indirectly referencing the values previously entered into the Points table (See Section 3.6.1.2 and Fig. 12), and (3) adding an additional value of 5. The indirectly referenced values corresponding to the indices of 1 and 2 are 10 and 12, respectively. Since the SUM toggle was selected, all of the values are simply added, yielding a final value of 27. The value associated with index 6 is obtained by (1) selecting the SUM – 1 toggle, (2) indirectly referencing the first point in the Points table, and (3) adding a value of 13. The value associated with index 7 is obtained by selecting the DIF + 1 toggle, and indirectly referencing the values associated with indices 6 and 1. The final table value, associated with index 8, is obtained by selecting the SUM toggle and simply entering a value in the TERMS column. This last operation is identical to the operation performed by the Points option (See Section 3.6.1.2). The final result of these entries are values that can be subsequently associated with segments that are generated from other segments through concatenation or extraction processes.

3.6.1.5 Composite Spacing

The Composite Spacing option is a more sophisticated version of the Spacing option described in Section 3.6.1.3. The screen depicted in Fig. 15 is displayed when the Composite Spacing option is invoked. Its operation is similar to that of the Composite Points function described in Section 3.6.1.4; however, this function operates on real numbers corresponding to spacings, rather than integer values. There are only three toggles available under this option; their definitions are identical to the corresponding toggles defined in Section 3.6.1.4. The operation of the Composite Spacing function is also identical to that of the Composite points function. The operations performed in the example of Fig. 15 are summarized as follows:

Toggle	Number	Operation	Result
SUM	5	Spacing(2) +	
		Spacing(3) +	
		0.4 =	
		0.02 +	
		0.1 +	
		0.4 =	0.52
DIF	6	Spacing(4) −	
		Spacing(2) −	
		Spacing(3) =	
		0.1 −	
		0.05 −	
		0.02 =	0.03
SUM	7	0.45	0.45
SUM	8	Spacing(2) +	
		0.5 =	
		0.02 +	
		0.5 =	0.52

3.6.2 Curves

This option allows the user to generate different types of curves. The curve menu options are depicted in Fig. 16. Each of the curve options is discussed in the following sections. Generally, all discussions of forms that generate geometric constructs will include the following information:

1. a description of the function of the form,
2. information that is required for proper operation of the form,
3. information that is optional,
4. notes pertaining to the proper operation of the form and to the structure of the resulting geometric construct, and
5. an example of the use of the form.

3.6.2.1 Line

This option provides the function to generate a straight line between two points, and to distribute an arbitrary number of points along the line according to specified spacing requirements. The screen shown in Fig. 17 is displayed when the Line option is selected.

Required Information

Endpoints:
X, Y, Z — Cartesian coordinates of endpoints, or

POINT — Index into previously defined list of points (See Section 3.6.1.1), or

SEGMENT — Identifier of previously defined segment, and
INDICES — Indices of a point on the segment (See Section 3.3)

Number of Points Along Line:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at Each Endpoint:
SPACING — See Section 3.3

File Output:
FILE — Output number of file (See Section 3.3)

Placement on Curved Surface:
CURVED SURFACE — Identifier of previously defined surface upon which the line will be constrained

Notes

If the line is to fall upon a curved segment, then the CURVED SURFACE must be entered first. The endpoints must be points on the curved surface.

Endpoints must be given prior to entering the number of points on the line.

The point progression along the line is from the first endpoint to the second endpoint.

Example

An example of the use of the Line function is depicted in Fig. 17. The first endpoint is point 3, which was defined using the Set Point function described in Section 3.6.1.1. The second endpoint is an absolute coordinate (0.0, 5.0, 0.0). The number of points along the line is defined indirectly by referring to the table of points generated by the Points function described in Section 3.6.1.2. Default spacing is used on the first endpoint, whereas the spacing on the second endpoint (0.02) is obtained indirectly by referring to the spacing table generated by the Spacing function described in Section 3.6.1.3. The straight line is stored as segment 1. Immediately after the DO IT button is selected with the mouse, the newly generated straight line is drawn in the graphics window in the lower left corner of the screen. A circle is displayed at the location of the first point on the line.

3.6.2.2 Cubic

This option provides the function to specify a cubic curve between two points, and to distribute an arbitrary number of points along the curve according to specified spacing requirements. The slope at each endpoint must be specified. The screen depicted in Fig. 18 is displayed when this option is selected.

Required Information

Endpoints:

X, Y, Z — Cartesian coordinates of endpoints, or

POINT — Index into previously defined list of points (See Section 3.6.1.1), or

SEGMENT — Identifier of previously defined segment, and
INDICES — Indices of a point on the segment (See Section 3.3)

Tangents at Endpoints:

UNIT TANGENT — Expressed as components in Cartesian space

Number of Points Along Curve:

POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at Each Endpoint:
SPACING — See Section 3.3

File Output:
FILE — Output number of file (See Section 3.3)

Placement on Curved Surface:
CURVED SURFACE — Identifier of previously defined surface upon which the curve will be constrained

Notes

If a line is to fall upon a curved segment then the CURVED SURFACE must be entered first. The endpoints must be points on the curved surface.

Endpoints must be given prior to entering the number of points on the line.

If an endpoint is to be defined as a reference number pointing to a pre-existing segment (i.e., by using the SEGMENT and INDICES option), the normal direction must be assigned prior to entering the segment number. If no unit tangent vector is defined for that endpoint, the cubic will leave the surface in the specified normal direction.

Point progression along the curve is from the first endpoint to the second endpoint.

Example

An example of the use of the Cubic function is depicted in Fig. 18. In this example, a cubic curve is adjoined to a previously generated straight line (See Section 3.6.2.1). The first endpoint of the cubic curve is defined as the last point (point 10) of the straight line (segment 1). The coordinates of the endpoint (0.0, 5.0, 0.0) are written in the appropriate location when the indirect reference is fully entered. The last endpoint is defined as point 2, previously defined using the Set Point function described in Section 3.6.1.1. The number of points to be distributed along the cubic curve is defined as 13. Unit tangents of the cubic curve are defined; the tangent at the first endpoint is specified to preserve slope continuity with the adjoining straight line. The spacing at the first endpoint is specified to be equal to the spacing at the last endpoint of the straight line. Default spacing is used at the second endpoint of the cubic curve. The result of this operation is displayed in the graphics window. A circle is displayed at the location of the first point on the curve.

3.6.2.3 Conic

The following options provide functions to generate conic sections in the X-Y plane. Conic sections include circles, ellipses, and circular, elliptic, parabolic, and hyperbolic arcs. The menu illustrated in Fig. 19 is displayed when the Conic option is selected.

3.6.2.3.1 Circle

This option allows a user to generate a circle (or portion of a circle) with the origin at (0,0,0). The screen depicted in Fig. 20 is displayed when this option is selected.

Required Information

Circle Radius:
RADIUS — Radius must be a positive nonzero value.

Angles at Endpoints:
ANGLE — Defaults are 0 and 360 deg for the first and last end angles, respectively.

Number of Points Along the Circle:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing at Endpoints:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

The number of points must be greater than or equal to three.

Angles are measured relative to the positive X axis.

Spacing at endpoints is relative to total subtended angle (i.e., last end angle minus first end angle).

Point progression is from the first end angle to the last end angle.

Example

In the example depicted in Fig. 20, 25 points are distributed along a circular arc extending from 0 to 180 deg. The number of points along the arc is described indirectly by referring to the table of values specified by the Points function described in Section 3.6.1.2. Default spacing, i.e., an equiangular distribution, is used.

3.6.2.3.2 Ellipse

This option allows a user to generate an ellipse (or portion of an ellipse). The screen depicted in Fig. 21 is displayed when this option is selected.

Required Information

Semi-Axes:
SEMI-AXIS — Must be a positive nonzero value

Angles at Endpoints:
ANGLE — Defaults are 0 and 360 deg for the first and last end angles, respectively.

Number of Points Along the Arc:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing at Endpoints:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

The number of points must be greater than or equal to three.
Angles are measured relative to the positive X axis.
Spacing at endpoints is relative to total angle (i.e., last end angle minus first end angle).
Point progression is from the first end angle to the last end angle.

Example

In the example depicted in Fig. 21, 25 points are distributed along an elliptical arc extending from 90 to 180 deg. The number of points along the arc is described indirectly by referring to the table of values specified by the Points function described in Section 3.6.1.2. Default spacing, i.e., an equiangular distribution, is used.

3.6.2.3.3 Circular Arc

This option allows a user to generate a circular arc. The Circle function may be used for the same purpose, but this option may be more convenient for applications where the desired X and Y intercepts are known. The screen depicted in Fig. 22 is displayed when this option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero number

Y-Intercept:
WIDTH — Positive nonzero number

Angles at Endpoints:

ANGLE — Defaults are −90 and 90 deg for the first and last end angles, respectively.

Number of Points Along the Arc:

POINTS — Actual number of points, or

NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:

SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing at Endpoints:

SPACING — Relative values, or

NUMBER — Index into spacing list (See Section 3.3)

File Output:

FILE — Output number of file (See Section 3.3)

Notes

The number of points must be greater than or equal to three.

Angles are measured relative to the positive X axis.

Spacing at endpoints is relative to total angle (i.e. last end angle minus first end angle).

Point progression is from the first end angle to the last end angle.

Example

In the example depicted in Fig. 22, 15 points are distributed along a circular arc between angles of −90 to 90 deg. The number of points along the arc is described indirectly by referring to the table of values specified by the Points function described in Section 3.6.1.2 . X and Y intercepts of 1.0 and 0.5, respectively, are specified. Default spacing is used for the second end, and a spacing of 0.15 is specified for the first end. Note that the origin of the coordinate axes does not in general correspond to the center of the circle.

3.6.2.3.4 Elliptical Arc

This option allows a user to generate an elliptical arc. This option is similar to the Ellipse function (See Section 3.6.2.3.2), but specifies the ellipse in terms of X and Y intercepts. The screen depicted in Fig. 23 is displayed when this option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero number

Y-Intercept:
WIDTH — Positive nonzero number

Angles at Endpoints:
ANGLE — Defaults are −90 and 90 deg for the first and last end angles, respectively.

Number of Points Along the Arc:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Eccentricity of Ellipse:
ECCENTRICITY — Number between 0 and 1

Optional Information

Spacing at Endpoints:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

The number of points must be greater than or equal to three.

ECCENTRICITY defaults to 0, in which case the operation of this function is theoretically identical to that of the CIRCULAR ARC function. In practice, a very small number must be input for ECCENTRICITY if a circular arc is desired.

Angles are measured relative to the positive X axis.

Spacing at endpoints is relative to total angle (i.e., last end angle minus first end angle).

Point progression is from the first end angle to the last end angle.

Example

In the example depicted in Fig. 23, 50 points are distributed along an elliptical arc between angles of −90 and 270 deg. X and Y intercepts of 1.0 are specified. An eccentricity of 0.9 is specified. Default spacing, i.e., an equiangular distribution, is used. Note that since the angular distribution is generated relative to the coordinate axes, the spacing around the elliptical arc varies. (See Fig. 41 for a clearer depiction of the equiangular distribution.)

3.6.2.3.5 Parabolic Arc

This option allows a user to generate a parabolic arc. The longitudinal axis of the parabola is the X axis. The screen depicted in Fig. 24 is displayed when this option is selected.

Required Information

X-Intercept:

LENGTH — Positive nonzero number

Y-Intercept:

WIDTH — Positive nonzero number

Angles at Endpoints:

ANGLE — Defaults are −90 and 90 deg for the first and last end angles, respectively.

Number of Points Along the Arc:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing at Endpoints:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

The number of points must be greater than or equal to three.
Angles are measured relative to the positive X axis.
Spacing at endpoints is relative to total angle (i.e., last end angle minus first end angle).
Point progression is from the first end angle to the last end angle.

Example

In the example depicted in Fig. 24, 25 points are distributed along a parabolic arc extending from −90 to 90 deg. The number of points along the arc is described indirectly by referring to the table of values specified by the Points function described in Section 3.6.1.2. X and Y intercept values of 1.0 and 0.5, respectively, are specified. Default spacing, i.e., an equiangular distribution, is used.

3.6.2.3.6 Hyperbolic Arc

This option allows a user to generate a hyperbolic arc, with the longitudinal axis along the X axis. The screen depicted in Fig. 25 is displayed when this option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero number

Y-intercept:
WIDTH — Positive nonzero number

Angles at Endpoints:
ANGLE — Defaults are −90 and 90 deg for the first and last end angles, respectively.

Number of Points Along the Arc:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Angle Between Asymptote and X-Axis:
ASYMPTOTE ANGLE — Positive real number

Optional Information

Spacing at Endpoints:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

The number of points must be greater than or equal to three.

Angles are measured relative to the positive X axis.

Spacing at endpoints is relative to total angle (i.e., last end angle minus first end angle).

Point progression is from the first end angle to the last end angle.

Example

In the example depicted in Fig. 25, 25 points are distributed along a hyperbolic arc extending from −90 to 90 deg. The number of points along the arc is described indirectly by referring to the table of values specified by the Points function described in Section 3.6.1.2. X and Y intercept values of 1.0 are specified. Default spacing, i.e. an equiangular distribution, is used. An asymptote angle of 145 deg is specified.

3.6.2.4 Distribution

This option allows the user to redistribute the spacing and number of points along a pre-existing curve. The screen shown in Fig. 26 is displayed when the Distribution option is selected.

Required Information

Curve to be Modified:
CURVE — Segment number of previously defined curve

Number of Points Along the Arc:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at Each Endpoint:
SPACING — See Section 3.3

File Output:
FILE — Output number of file (See Section 3.3)

Pack Points in Regions of High Curvature:
CURVATURE SPACING — Selection of this toggle will cause points to be more closely packed in regions of high curvature.

Example

The operation of the Distribution function is illustrated in Fig. 26. A previously generated curve, segment 11, has been accessed for processing. Segment 11 was generated by distributing a small number of points along a hyperbolic curve. The distribution of points along the hyperbolic curve is such that very poor resolution is evident. In this example, a new curve, segment 14, is generated from the points along the hyperbolic curve. The new curve contains 35 points. A relative spacing of 0.1 is taken from the previously generated Spacing table (See Section 3.6.1.3) and is applied to the first endpoint of the new curve. A relative spacing of 0.135 is applied to the last end. The new curve is a cubic spline; as such, it will not, in general, preserve the characteristics of the curve that generated the points through which the spline passes.

3.6.2.5 Composite Curve

This option allows the user to generate a curve by combining a list or range of pre-existing curves. The screen shown in Fig. 27 is displayed when the Composite Curve option is selected.

Required Information

Segments to be Combined:
SEGMENT—Segment numbers of previously defined curves

Output Segment Identifier:
COMPOSITE SEGMENT — Segment number (See Section 3.3)

Optional Information

File Output:
FILE — Output number of file (See Section 3.3)

Example

The previously generated straight line and cubic curves (See Sections 3.6.2.1 and 3.6.2.2) are combined into segment 20 in the example depicted in Fig. 27. The newly defined segment is displayed in the graphics window. The new segment preserves the spacing distributions that were imposed on the straight line and cubic curve.

3.6.2.6 Patch Curve

This option allows the user to generate a cubic curve by connecting two points, each of which is contained in pre-existing curve segments. The screen shown in Fig. 28 is displayed when the Patch Curve option is selected.

Required Information

Curves That Will Supply Endpoints:
FIRST CURVE, LAST CURVE — Segment identifier of previously defined curves

Points on the Curves That Will be Endpoints of the New Curve:
POINT — Integers between 1 and the total number of points in the appropriate curve, inclusive

Directions of Endpoints of New Curve. Toggle Boxes Are Used to Select the Six Available Directions:
+TAN — positive tangent
−TAN — negative tangent
+NOR — positive normal
−NOR — negative normal
+BIN — positive binormal
−BIN — negative binormal

Number of Points Along the Arc:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Arc Length of New Curve:
TOTARC — Total arc length, positive real number

Optional Information

Spacing information at Each Endpoint:
SPACING — See Section 3.3

File Output:

FILE — Output number of file (See Section 3.3)

Placement on Curved Surface:

CURVED SURFACE — Identifier of previously defined surface upon which the curve will be constrained

Notes

All information pertaining to the first end of the patch curve should be entered before proceeding to the second end.

Selection of an endpoint direction is accomplished by placing the cursor on top of the desired direction box.

If the resulting segment is to fall upon a curved segment, then the CURVED SURFACE identifier must be entered first.

Example

The use of the Patch operation is depicted in Fig. 28. A cubic curve consisting of 25 points is generated between the previously generated straight line and cubic curve (See Sections 3.6.2.1 and 3.6.2.2). The cubic curve progresses from the third point on the straight line (segment 1) to the eleventh point on the cubic curve (segment 2). The curve is required to be tangent to segment 1, and normal to segment 2. Relative spacings of 0.02 are imposed at each endpoint, and a total arc length of 10.0 is imposed. The resulting segment is displayed and highlighted in the graphics area.

3.6.2.7 Intersection

This option allows the user to generate a curve segment by intersecting two pre-existing surface segments. The screen shown in Fig. 29 is displayed when the Intersection option is selected.

Required Information

Intersected Surface:

FEMALE SEGMENT — Segment number of a previously defined surface

Intersecting Surface:

MALE SEGMENT — Segment number of a previously defined surface that is intersecting the female surface

Optional Information

File Output:

FILE — Output number of file (See Section 3.3)

Notes

The male segment number must be entered first.

The male segment number must have its slower-running (See Section 3.3) direction crossing the female segment.

The entire screen of data must be re-entered after an error has occurred or the screen has been processed.

Example

An elliptical cylinder and a planar surface, both previously generated, intersect in the example depicted in Fig. 29. The cylinder is the male segment (segment 4), and the planar surface is the female segment (segment 7). The intersection that results is a curve consisting of points along the male segment's slower-running coordinate that intersects the female segment. In this case, the intersection is an ellipse.

3.6.3 Surfaces

This option provides functions that generate a variety of surfaces. The surface menu options are depicted in Fig. 30, and are discussed in the following sections.

3.6.3.1 Rotate

This option allows the user to generate a surface by rotating a previously defined curve segment about a specified axis. Alternatively, two previously defined curve segments may be rotated about a specified axis, and a surface may be generated between the rotated curve segments. The screen shown in Fig. 31 is displayed when the Rotate option is selected.

Required Information

Bounding Curve:
FIRST BOUNDING CURVE — Segment identifier of the curve that will be the boundary at the initial rotational angle. If only FIRST BOUNDING CURVE is specified, the curve will also serve as LAST BOUNDING CURVE (See Optional Information).

Initial and Final Rotational Angles:
ANGLE — Angles of rotation, in degrees, from the initial curve position

Number of Curves on Generated Surface:
ANGLES — The actual number of rotated curves on the surface, or
NUMBER — An index into a list of numbers (See Section 3.3)

Axis of Rotation:
ROTATION AXIS — Direction cosines of axis of rotation

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing at Boundaries:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

Interpolation (Used Only if Two Bounding Curves Are Specified):
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3), or
FUNCTION — See example, Section 3.6.3.3

File Output:
FILE — Output number of file (See Section 3.3)

Bounding Curve:
LAST BOUNDING CURVE — Segment identifier of the curve that will be the boundary at the final rotational angle

Note

FIRST BOUNDING CURVE must be entered first.

Example

The use of the Rotate function is depicted in Fig. 31. The composite curve generated in the Example of Section 3.6.2.5 is rotated about the X axis. Since no LAST BOUNDING CURVE is specified, the composite curve serves as both boundaries of the surface. Fifteen circumferential locations are specified from 0 to 180 deg. Default spacing is selected for this example; therefore, the circumferential spacing is equiangular. The resulting surface of rotation is depicted in the graphics area.

3.6.3.2 Stack

This option allows the user to generate a surface by stacking a predefined curve along a specified axis or curve. The screen shown in Fig. 32 is displayed when the Stack option is selected.

Required Information

Bounding Curve:
FIRST BOUNDING CURVE — Segment identifier of the curve that will be the initial boundary. If only FIRST BOUNDING CURVE is specified, the curve will also serve as LAST BOUNDING CURVE (See Optional Information).

Axis Tangent and Normal:
AXIS TANGENT — Tangent vector of axis system in which bounding curves are defined
AXIS NORMAL — Principal normal vector of axis system in which bounding curves are defined

Stacking Axis:
AXIS — Segment identifier along which bounding curves will be stacked

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)
NORMAL POINT — A point in space that defines a normal direction (used if axis is a straight line)

Optional Information

Spacing at Boundaries:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

Interpolation (Used Only if Two Bounding Curves Are Specified):
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3), or
FUNCTION — See example, Section 3.6.3.3

File Output:
FILE — Output number of file (See Section 3.3)

Bounding Curve:
LAST BOUNDING CURVE — Segment identifier of the curve that will be blended with FIRST BOUNDING CURVE

Notes

FIRST BOUNDING CURVE must be the first input.
Tangent and normal vectors may be input in degrees or as direction cosines.

Example

The use of the Stack function is illustrated in Fig. 32. Previously generated curve segments are used in this example. The segment specified as the first bounding curve (segment 24) is an ellipse generated by the conic curve function described in Section 3.6.2.3.2. The ellipse is depicted in the graphics window. The axis (segment 4) is a circular arc also generated by the appropriate conic curve generation function (See Section 3.6.2.3.3). The values assigned to AXIS TANGENT state that the Z axis of the ellipse (i.e. out of the screen) is to be parallel to the local tangent of the axis. The values entered under AXIS NORMAL specify that the X axis of the ellipse is to be aligned with the normal direction of the axis. The normal direction of the axis is the direction of maximum curvature; if the axis is a straight line (i.e. has zero curvature), the normal direction must be specified by entering a normal point. The result of the stacking operation is depicted in the graphics area.

3.6.3.3 Blend

This option allows the user to generate a surface by blending one curve into another. The screen shown in Figs. 33 and 34 is displayed when the Blend option is selected.

<u>**Required Information**</u>

Bounding Curves:
FIRST BOUNDING CURVE — Segment identifier of curve that will form the first boundary of resulting surface
LAST BOUNDING CURVE — Segment identifier of curve that will form the last boundary of resulting surface

Number of Curves:
CURVES — The actual number of curves to be generated on the surface by the blending function, or
NUMBER — An index into a table of numbers (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

<u>**Optional Information**</u>

Interpolation:
SPACING — Relative spacings at first and last boundary of generated surface, or
NUMBER — Index into list of spacings (See Section 3.2), or
FUNCTION — Previously defined curve segment that is normalized to yield the interpolation values (See Example)

File Output:
FILE — Output number of file (See Section 3.3)

Placement on Curved Surface:
CURVED SURFACE — Identifier of previously defined surface upon which the generated surface will be constrained

<u>**Note**</u>

If line is to fall upon a curved segment, then the CURVED SURFACE box must be filled first.

Example

The use of the Blend function is illustrated in Figs. 33 and 34. In Fig. 33, two cubic curves (segments 1 and 2) are specified as the boundaries. Fifteen curves are specified along the slower-running coordinate of the surface. Relative spacings of 0.001 and 0.005 are specified at the upper and lower boundaries, respectively. The result of the blending operation is displayed in the graphics area.

The Rotate, Stack, and Blend operations allow spacing to be specified by defining a monotonic function that controls spacing over the slower-running coordinate. In Fig. 34, two horizontal straight lines are blended together, according to a function (segment 6), that was generated by concatenating two cubic curves. (This function is plotted in the graphics window in this example.) The function is normalized; as a result, only the shape of the function determines the spacing on the blended surface. The number of points, the physical dimensions, and the point distribution on the curve defining the function are irrelevant. However, the function must be monotonic to guarantee a unique mapping.

3.6.3.4 Transur

This option allows the user to generate a surface by transfinite interpolation. This option requires the previous definition of four bounding curves. The number of points and the spacing along each coordinate direction of the resulting surface are defined by the bounding surfaces. The screen shown in Fig. 35 is displayed when the Transur option is selected.

Required Information

Bounding Curves:
LOWER1 EDGE CURVE, UPPER1 EDGE CURVE, LOWER2 EDGE CURVE, UPPER2 EDGE CURVE — Segment identifiers of previously generated curves

Output Segment identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

File Output:
FILE — Output number of file (See Section 3.3)

Placement on Curved Surface:

CURVED SURFACE — Identifier of previously defined surface upon which the generated surface will be constrained

Notes

If line is to fall upon a curved segment, then the CURVED SURFACE identifier must be entered first.

Point progressions and number of points on curves bounding opposite sides of the generated surface should be the same, or strange results will occur.

Example

In the example depicted in Fig. 35, the two cubic curves used in the Blend example (segments 1 and 2; see Section 3.6.3.3) and two vertical lines (segments 4 and 5) connecting the endpoints of the cubic curves are used as boundaries for the interpolation. The result of the transfinite interpolation is displayed in the graphics area.

3.6.3.5 Patch

This option allows the user to generate a cubic surface bounded by curves extracted from pre-existing surfaces. The slopes at the boundaries of the cubic surface can be specified to be normal or tangent to the surfaces from which the bounding curves were extracted. The screen shown in Fig. 36 is displayed when the Patch option is selected.

Required Information

Surfaces That Will Supply Boundaries to the Resulting Cubic Surface:

FIRST SURFACE, LAST SURFACE — Segment numbers of previously defined surfaces (See Section 3.3)

Number of Points on the Resulting Cubic Surface, in the Coordinate Direction Spanning the Two Boundary Surfaces:

POINTS — Actual number of points, or

NUMBER — Index into numbers list (See Section 3.3)

Points on the Surfaces That Define the Endpoints of a Boundary Line:

FIRST POINT, LAST POINT — Indices of points on surface. The endpoints must define a pre-existing line on the surface (See Section 3.3)

Edges of Surfaces That Define Boundary Lines:

UPPER1, UPPER2, LOWER1, LOWER2 — Toggles that will select the entire edge as a boundary line

Slope of the Cubic Surface at the Boundary Lines:

+NOR, −NOR — Positive and negative normals

+TAN, −TAN — Positive and negative tangents

Output Segment Identifier:

SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at the Bounding Surfaces:

SPACING — See Section 3.3

File Output:

FILE — Output number of file (See Section 3.3)

Placement on Curved Surface:

CURVED SURFACE — Identifier of previously defined surface upon which the generated surface will be constrained

Arc Length of Each Curve Comprising the New Surface:

TOTARC — Total arc length, positive real number

Notes

All information pertaining to the first end of the cubic patch should be entered before proceeding to the second end.

Spacing and total arc length are required if +NOR or −NOR is specified.

If +TAN or −TAN is selected, default spacing will be the spacing of the boundary surfaces at the line of intersection between the cubic patch and the boundary surfaces.

Example

In the example illustrated in Fig. 36, two short curve segments are defined on previously generated horizontal surfaces. The resulting cubic patch curve is required to be tangent to the lower surface and normal to the upper surface. Twenty points are specified along the slower-running surface coordinate. Relative spacings of 0.01 are specified at the ends of the cubic patch. A total arc length of 20 is specified. The result of the cubic patch operation is displayed in the graphics area.

3.6.3.6 Flat Conic

This option allows the user to generate a conic surface in the X-Y plane (i.e. Z = 0). The surface may be a closed circle or ellipse, or it may be an arc segment of a circle, ellipse, parabola, or hyperbola. The user may choose which type of surface to generate through a menu shown in Fig. 37. The individual forms are very similar to those used to generate conic curves (See Section 3.6.2.3); the only additional information necessary for the following surface generation functions is the number of points desired in the radial coordinate direction. Each of the surface types will be discussed in the following sections.

3.6.3.6.1 Circle

This option allows the user to generate a circular surface with its origin at (0,0,0). The screen shown in Fig. 38 is displayed when the Circle option is selected.

Required Information

Circle Radius:
RADIUS — Positive nonzero real number

Angles at Circumferential Boundaries:
ANGLE — Real numbers

Number of Points in the Circumferential Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points in the Radial Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information in the Circumferential Direction:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Note

The user must specify three or more points in each of the radial and circumferential directions.

Example

In the example depicted in Fig. 38, a circular surface consisting of 25 circumferential and 15 radial points is generated. The surface extends between angles of 0 to 180 deg. The numbers of points in the circumferential and radial directions are obtained by indirectly referring to the table of values built in the example of Section 3.6.1.2. Default spacing is used in both directions. The result of the surface generation process is displayed in the graphics area.

3.6.3.6.2 Ellipse

This option allows the user to generate an elliptical surface. The screen shown in Fig. 39 is displayed when the Ellipse option is selected.

Required Information

Semi-Axes:
SEMI-AXIS — Positive real number

Starting and Ending Angles in the Circumferential Direction:
ANGLE — Real numbers

Number of Points in the Circumferential Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points in the Radial Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at Endpoints (Circumferential Direction):
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Note

The user must specify three or more points in each of the radial and circumferential directions.

Example

The generation of an elliptical surface is illustrated in Fig. 39. The surface extends between angles of 90 and 180 deg, and consists of 25 circumferential and 15 radial points. The numbers of radial and circumferential points are obtained indirectly by referring to the previously generated Points table (See Section 3.6.1.2). Default spacing is used in both surface coordinate directions. The result of the surface generation process is displayed in the graphics area.

3.6.3.6.3 Circular Arc

This option allows a user to generate a circular surface. The Circle surface generation function may be used for the same purpose, but the method used in this option may be more convenient for certain applications where X- and Y-intercepts are known. The screen depicted in Fig. 40 is displayed when this option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero real number

Y-Intercept:
WIDTH — Positive nonzero real number

Angles at Circumferential Boundaries:
ANGLE — Real numbers

Number of Points in the Circumferential Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points in the Radial Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at Endpoints in the Circumferential Direction:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Note

The user must specify three or more points in each of the circumferential and radial directions.

Example

In the example depicted in Fig. 40, a circular surface consisting of 15 circumferential and 15 radial points is generated. The surface extends between angles of −90 and 90 deg. X- and Y-intercept values of 1.0 and 0.5, respectively, are specified. The numbers of points in the circumferential and radial directions are obtained by indirectly referring to the table of values built in the example of Section 3.6.1.2. Default spacing is used in both directions. The result of the surface generation process is displayed in the graphics area.

3.6.3.6.4 Elliptical Arc

This option allows the user to generate a surface with an elliptical boundary, where the major axis lies on the X axis. This option performs nearly the same operation as the ELLIPSE function, but defines the ellipse in terms of X- and Y-intercepts. The screen shown in Fig. 41 is displayed when the Elliptical Arc option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero real number

Y-Intercept:
WIDTH — Positive nonzero real number

Angles at Endpoints in Circumferential Direction:
ANGLE — Real numbers

Number of Points in the Circumferential Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points in the Radial Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Eccentricity of Ellipse:
ECCENTRICITY — Positive real number between 0 and 1

Optional Information

Spacing Information at Endpoints in the Circumferential Direction:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

The user must specify three or more points in each of the circumferential and radial directions.

ECCENTRICITY defaults to 0, in which case the operation of this function is theoretically identical to that of the CIRCULAR ARC function. In reality, ECCENTRICITY must be defined as a small nonzero number in order to approximate a circular arc.

Example

The generation of an elliptical surface is illustrated in Fig. 41. The surface extends between angles of −90 to 270 deg, and consists of 50 circumferential and 15 radial points. X- and Y-intercept values of 1.0 are specified. The number of radial points is obtained indirectly by referring to a previously generated Points table (See example, Section 3.6.1.2), whereas the number of circumferential points is specified directly. Default spacing is used in both surface coordinate directions. The eccentricity is 0.9. The result of the surface generation process is displayed in the graphics area.

3.6.3.6.5 Parabolic Arc

This option allows the user to generate a surface with a parabolic boundary. The screen shown in Fig. 42 is displayed when the Parabolic Arc option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero real number

Y-Intercept:
WIDTH — Positive nonzero real number

Angles at Endpoints in Circumferential Direction:
ANGLE — Real numbers

Number of Points in the Circumferential Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points in the Radial Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information at Endpoints in the Circumferential Direction:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Note

The user must specify three or more points in each of the circumferential and radial directions.

Example

The generation of a surface using the Parabolic Arc function is illustrated in Fig. 42. The surface extends between −90 and 90 deg in the circumferential direction. X- and Y-intercepts of 1.0 and 0.5, respectively, are specified. Fifteen radial and 25 circumferential points are specified. The numbers of radial and circumferential points are obtained indirectly by referring to a previously generated Points table (See example, Section 3.6.1.2). The result of the surface generation process is displayed in the graphics area.

3.6.3.6.6 Hyperbolic Arc

This option allows the user to generate a surface where one boundary is a hyperbolic arc. The screen shown in Fig. 43 is displayed when the Hyperbolic option is selected.

Required Information

X-Intercept:
LENGTH — Positive nonzero real number

Y-Intercept:
WIDTH — Positive nonzero real number

Angles at Endpoints Relative to Origin:
ANGLE — Real numbers

Number of Points in the Circumferential Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points in the Radial Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Angle Between Asymptote and X Axis:
ASYMPTOTE ANGLE — Positive real number

Optional Information

Spacing Information at Endpoints, in Circumferential Direction:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Note

The user must specify three or more points in each of the circumferential and radial directions.

Example

The generation of a surface using the Hyperbolic Arc function is illustrated in Fig. 43. The surface extends between −90 and 90 deg in the circumferential direction. X- and Y-intercepts of 1.0 are specified. Fifteen radial and 25 circumferential points are specified. The numbers of radial and circumferential points are obtained indirectly by referring to a previously generated Points table (See example, Section 3.6.1.2). An asymptote angle of 145.0 deg is specified. The result of the surface generation process is displayed in the graphics area.

3.6.3.7 Conic

This option allows the user to generate conic-section surfaces, which may be a segment of a sphere, ellipsoid, elliptical cone, or elliptical paraboloid. The user may choose the type of surface to generate through the menu depicted in Fig. 44. Each of the surface types will be discussed in the following sections.

3.6.3.7.1 Sphere

This option allows the user to generate a spherical surface whose origin is at (0,0,0). The screen shown in Fig. 45 is displayed when the Sphere option is selected.

Required Information

Radius of Sphere:
RADIUS — Positive real number

Starting and Ending Latitude:
LAT1 — Starting latitude, measured from positive Z axis, in degrees
LAT2 — Ending latitude

Starting and Ending Longitude:
LON1 — Starting longitude, measured from X-Z plane, in degrees
LON2 — Ending longitude

Number of Points (Latitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points (Longitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information for All Sides of Surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

Default starting and ending values for longitude are 0.0 and 360.0 deg.
Default starting and ending values for latitude are 0.0 and 180.0 deg.

Example

The use of the Sphere function is illustrated in Fig. 45. A spherical surface is generated between latitudes of 0 and 180 deg, and between longitudes of 0 and 90 deg. Twenty-five points are specified in the latitudinal direction, and 15 in the longitudinal direction. The numbers of points in each coordinate direction are obtained by referring to a previously generated Points table (See example, Section 3.6.1.2). Default spacings are used at all edges of the surface. The result of the surface generation process is displayed in the graphics area.

3.6.3.7.2 Ellipsoid

This option allows the user to generate an ellipsoidal surface. The screen shown in Fig. 46 is displayed when the Ellipsoid option is selected. The screen is very similar to the Sphere form (See Section 3.6.3.7.1), but requires semi-axes to be defined instead of a radius.

Required Information

Semi-Axes of Ellipsoid:
SEMIAX — Positive real number for each Cartesian direction

Starting and Ending Latitude:
LAT1 — Starting latitude, measured from positive Z axis, in degrees
LAT2 — Ending latitude

Starting and Ending Longitude:
LON1 — Starting longitude, measured from X-Z plane, in degrees
LON2 — Ending longitude

Number of Points (Latitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points (Longitude)
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information for All Sides of Surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

Default starting and ending values for longitude are 0.0 and 360.0 deg.
Default starting and ending values for latitude are 0.0 and 180.0 deg.

Example

The generation of an ellipsoidal surface is illustrated in Fig. 46. The surface is generated between latitudes of 0 and 180 deg, and between longitudes of 0 and 90 deg. Twenty-five points are specified in the latitudinal direction, and 15 in the longitudinal direction. The numbers of points in each coordinate direction are obtained by referring to a previously generated Points table (See example, Section 3.6.1.2). Semi-axis values of 8.0, 2.0, and 4.0 are specified for the X, Y, and Z directions, respectively. Default spacings are used at all edges of the surface. The result of the surface generation process is displayed in the graphics area.

3.6.3.7.3 Spherical Arc

This option allows the user to generate a spheroidal surface. The screen shown in Fig. 47 is displayed when the Spherical Arc option is selected. The screen is very similar to the Sphere form (See Section 3.6.3.7.1), but requires three distances from the origin to the spheroidal surface to be defined instead of a radius.

Required Information

Z-Intercept:
LENGTH — Positive nonzero real number

X- and Y-Intercepts:
WIDTH — Positive nonzero real number

Starting and Ending Latitude:
LAT1 — Starting latitude, measured from positive Z axis, in degrees
LAT2 — Ending latitude

Starting and Ending Longitude:
LON1 — Starting longitude, measured from X-Z plane, in degrees
LON2 — Ending longitude

Number of Points (Latitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points (Longitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information for All Sides of Surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

Default starting and ending values for longitude are 0.0 and 360.0 deg.
Default starting and ending values for latitude are 0.0 and 90.0 deg.

Example

The use of the Spherical Arc function is illustrated in Fig. 47. A spherical surface is generated between latitudes and longitudes of 0 and 90 deg. Twenty-five points are specified in the latitudinal direction, and 15 in the longitudinal direction. The numbers of points in each coordinate direction are obtained by referring to a previously generated Points table (See example, Section 3.6.1.2). X-, Y-, and Z-intercept values of 4.0, 2.0, and 5.0 are specified, respectively. Default spacings are used at all edges of the surface. The result of the surface generation process is displayed in the graphics area.

3.6.3.7.4 Ellipsoidal Arc

This option generates an ellipsoidal surface. The screen shown in Fig. 48 is displayed when the Ellipsoidal Arc option is selected. The screen is very similar to the Ellipsoid (See Section 3.6.3.7.2) form, but requires three intercepts and an eccentricity to be defined, instead of the semi-axes.

Required Information

Z-Intercept:
LENGTH — Positive nonzero real number

X- and Y-Intercepts:
WIDTH — Positive nonzero real number

Starting and Ending Latitude:
LAT1 — Starting latitude, measured from positive Z axis, in degrees
LAT2 — Ending latitude

Starting and Ending Longitude:
LON1 — Starting longitude, measured from X-Z plane, in degrees
LON2 — Ending longitude

Number of Points (Latitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points (Longitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Eccentricity:
ECCENTRICITY — Positive real number between 0 and 1

Optional Information

Spacing Information for All Sides of Surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

Default starting and ending values for longitude are 0.0 and 360.0 deg.
Default starting and ending values for latitude are 0.0 and 90.0 deg.
ECCENTRICITY defaults to zero, in which case the Ellipsoidal Arc function is in principle identical to the Spherical Arc function. In practice, ECCENTRICITY must be specified as a small positive number to approximate the Spherical Arc function.

Example

The use of the Ellipsoidal Arc function is illustrated in Fig. 48. A surface is generated between latitudes of 0 and 90 deg, and longitudes of 0 and 180 deg. Twenty-five points are specified in the latitudinal direction, and 15 in the longitudinal direction. The numbers of points in each coordinate direction are obtained by referring to a previously generated Points table (See example, Section 3.6.1.2). X-, Y-, and Z-intercept values of 2.0, 2.0, and 5.0 are specified, respectively. Default spacings are used at all edges of the surface. The eccentricity is 0.95. The result of the surface generation process is displayed in the graphics area.

3.6.3.7.5 Elliptical Cone

This option allows the user to generate an elliptical conic surface. (An elliptical cone has an elliptical cross-section). The screen shown in Fig. 49 is displayed when the Elliptical Cone option is selected.

Required Information

Z-Intercept:
LENGTH — Positive nonzero real number

X- and Y-Intercepts:
WIDTH — Positive nonzero real number

Starting and Ending Latitude:
LAT1 — Starting latitude, measured from positive Z axis, in degrees
LAT2 — Ending latitude

Starting and Ending Longitude:
LON1 — Starting longitude, measured from X-Z plane, in degrees
LON2 — Ending longitude

Number of Points (Latitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points (Longitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing information for all sides of surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File output:
FILE — Output number of file (See Section 3.3)

Notes

Default starting and ending values for longitude are 0.0 and 360.0 deg.
Default starting and ending values for latitude are 0.0 and 90.0 deg.

Example

The use of the Elliptical Cone function is illustrated in Fig. 49. A surface is generated between latitudes and longitudes of 0 and 90 deg. Twenty-five points are specified in the latitudinal direction, and 15 in the longitudinal direction. The numbers of points in each coordinate direction are obtained by referring to a previously generated Points table (See example, Section 3.6.1.2). X-, Y-, and Z- intercept values of 1.0, 2.0, and 5.0 are specified, respectively. Default spacings are used at all edges of the surface. The result of the surface generation process is displayed in the graphics area.

3.6.3.7.6 Elliptic Parabolic

This option allows the user to generate a surface on an elliptic paraboloid. The screen shown in Fig. 50 is displayed when the Elliptical Parabolic option is selected.

Required Information

Z-Intercept:
LENGTH — Positive nonzero real number

X- and Y-Intercept:
WIDTH — Positive nonzero real number

Starting and Ending Latitude:
LAT1 — Starting latitude, measured from positive Z axis, in degrees
LAT2 — Ending latitude

Starting and Ending Longitude:
LON1 — Starting longitude, measured from X-Z plane, in degrees
LON2 — Ending longitude

Number of Points (Latitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Number of Points (Longitude):
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Spacing Information for All Sides of Surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

File Output:
FILE — Output number of file (See Section 3.3)

Notes

Default starting and ending values for longitude are 0.0 and 360.0 deg.
Default starting and ending values for latitude are 0.0 and 90.0 deg.

Example

The use of the Elliptic Paraboloid function is illustrated in Fig. 50. A surface is generated between latitudes and longitudes of 0 and 90 deg. Twenty-five points are specified in the latitudinal direction, and 15 in the longitudinal direction. The numbers of points in each coordinate direction are obtained by referring to a previously generated Points table (See example, Section 3.6.1.2). X-, Y-, and Z- intercept values of 2.0, 1.0, and 5.0 are specified, respectively. Default spacings are used at all edges of the surface. The result of the surface generation process is displayed in the graphics area.

3.6.3.8 Surface Distribution

This option allows the user to redistribute the spacing and number of points along a pre-existing surface. The screen shown in Fig. 51 is displayed when the Surface Distribution option is selected.

Required Information

Surface to be Modified:
SURFACE — Surface identifier of previously generated surface

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Numbers of Points in Each Surface Coordinate Direction:
POINTS — Actual number of points, or
NUMBER — Index into numbers list (See Section 3.3)

Optional Information

Spacing Information for All Sides of Surface:
SPACING — Relative values, or
NUMBER — Index into spacing list (See Section 3.3)

Example

The use of the Surface Distribution function is illustrated in Fig. 51. The surface generated in the example of the use of the Transfinite Interpolation function (See Section 3.6.3.4) is used as input to this function. The numbers of points in each coordinate direction are increased relative to the example of Section 3.6.3.4. Relative spacings of 0.001 are specified at the upper and lower boundaries. A relative spacing of 0.05 is specified at the left boundary; default spacing is used at the right boundary. The resulting surface distribution is displayed in the graphics area.

3.6.4 Utilities

This option allows the user to perform various operations on existing segments. The user may choose the desired operation through the menu depicted in Fig. 52. Each of the operations on the submenu is discussed in the following sections.

3.6.4.1 Insert

This option allows the insertion of one segment into another at a specified starting position. This option is valid for both curve and surface segments. The screen shown in Fig. 53 is displayed when the Insert option is selected.

Required Information

Segment to Insert:
INSERTED SEGMENT — Segment number

Segment in Which to Insert Segment:
OLD SEGMENT — Segment number

Point in Old Segment Where New Segment Is to Be Inserted:
STARTING POINT — Point indices

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

File Output:
FILE — Output number of file (See Section 3.3)

Note

Segment number to insert must be entered first.

Example

The use of the Insert function is illustrated in Fig. 53. In this example, a rectangular surface element (designated as segment 18) is appended to the surface generated in the example of the use of the Surface Distribution function (segment 9, See Section 3.6.3.8). The right boundary of segment 9 must be overwritten with the left boundary of segment 18; since the faster-running coordinate of segment 9 is 30 points in length, the starting point is (30,1). The result of the insertion operation is highlighted and displayed in the graphics area.

3.6.4.2 Extract

This option allows the user to extract a segment from another segment at a specified starting position. This option is valid for both curve and surface segments. The screen shown in Fig. 54 is displayed when the Extract option is selected.

Required Information

Segment from Which New Segment Is to Be Extracted:
SEGMENT — Segment number

Point in Segment from Which New Segment Is to Be Extracted:
STARTING POINT — Point indices

Dimensions of New Segment:
POINTS — Point indices

New Segment Identifier:
EXTRACTED SEGMENT — Segment number (See Section 3.3)

Optional Information

File Output:
FILE — Output number of file (See Section 3.3)

Note

Old segment number must be entered first.

Example

The use of the Extract function is depicted in Fig. 54. The surface segment generated in the example of the use of the Blend function (See Section 3.6.3.3) is used as the segment from which another segment is to be extracted. A starting point of (6,1) is selected. Ten points along the faster-running coordinate and 9 points along the slower-running coordinate are designated as the extent of the extracted element. The extracted element is displayed and highlighted in the graphics area.

3.6.4.3 Trans

This option allows the user to translate, rotate, and scale a single segment, up to seven separate segments, or a range of segments. The screen shown in Fig. 55 is displayed when the Trans option is selected.

Required Information

Segments to Be Transformed:
SEGMENTS — Segment numbers. Up to seven segments can be specified. A range of segments can be specified by entering the starting and ending segment numbers in consecutive boxes, and by selecting the INCLUSIVE box.

Optional Information

Output Segments:

TRANSFORMED SEGMENTS — Segment numbers. Arbitrary numbers can be placed in these boxes to designate the segment identifiers of the transformed segments. A range of output segments can be designated in the same manner as ranges are defined for input (See Required Information). If no output segment identifiers are entered, the transformed segments will be written over the input segments.

Scaling Information:

SCALE FACTORS — The transformed segments will be scaled in X, Y, and Z by the numbers input in the appropriate boxes. Default scaling is 1.0, 1.0, 1.0.

Translation Information:

ORIGIN — The origin will be translated according to values entered in these boxes. Default translation is 0.0, 0.0, 0.0.

Rotation Information:

X UNIT VECTOR — Direction cosines of unrotated X axis, relative to the rotated coordinate system. Default is 1.0, 0.0, 0.0.

Y UNIT VECTOR — Direction cosines of unrotated Y axis, relative to the rotated coordinate system. Default is 0.0, 1.0, 0.0.

Z UNIT VECTOR — Direction cosines of unrotated Z axis, relative to the rotated coordinate system. Default is 0.0, 0.0, 1.0.

Notes

The Trans function performs scaling, rotation, and translation in that order.

Mirror images of segments may be created by using negative scale factors.

Projections of segments onto coordinate planes may be created by using zero-scale factors. For instance, the projection of a segment onto the Z = 0 plane may be obtained by using scale factors of 1.0, 1.0, 0.0.

The Trans function may be used to copy segments by using default scaling, rotation, and translation values.

Example

The use of the Trans function is depicted in Fig. 55. The surface segment generated in the example of the use of the Extract function (See Section 3.6.4.2) is translated, rotated, and scaled. The segment is translated along the X axis, rotated about the Z axis, and then scaled along the new X axis. The translated, rotated, and scaled segment is displayed and highlighted in the graphics area.

3.6.4.4 Parametrics

This operation maps a physical segment into a two-dimensional computational space. The segment must lie on a previously defined surface that will define the parameters of the computational space. The screen shown in Fig. 56 is displayed when the Parametrics option is selected.

Required Information

Surface Defining Parameter Space:
SURFACE — Segment number of previously defined surface

Segment to Be Parametrized:
SEGMENT — Segment number of previously defined curve or surface segment

Output Segment Identifier:
LATTICE — Segment number (See Section 3.3)

Optional Information

File Output:
FILE — Output number of file (See Section 3.3)

Example

The use of the Parametrics function is illustrated in Fig. 56. A circle generated with the Circle option (See Section 3.6.3.2.1) is placed on the two-dimensional surface generated in the example of the use of the Insert function (See Section 3.6.4.1). The mapping operation creates the elongated curve segment (segment 24) displayed in the upper right corner of the graphics area. The elongated curve is the circle plotted in parametric coordinates defined by the two-dimensional surface.

3.6.4.5 Lattice

This option performs a mapping from computational space to physical space, and hence is essentially the reverse of the Parametric option (See Section 3.6.4.4). A curve or surface segment defined in computational coordinates may be mapped onto a surface defined in physical space. The screen shown in Fig. 57 is displayed when the Lattice option is selected.

Required Information

Surface Defined in Physical Space:
SURFACE — Segment number of previously defined surface

Segment Defined in Computational Coordinates to be Mapped Onto Physical Surface:
LATTICE — Segment number of previously defined curve or surface segment

New Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

File Output:
FILE — Output number of file (See Section 3.3)

Example

The use of the Lattice function is illustrated in Fig. 57. A previously generated circle (not shown, but designated as segment 23) is mapped onto the surface generated in the example of the use of the Insert function (See Section 3.6.4.1). The resulting mapped segment (segment 20) is highlighted and displayed in the graphics area.

3.6.4.6 Switch

This operation modifies the point progression on a previously defined curve or surface segment. The screen shown in Fig. 58 is displayed when the Switch option is selected.

Required Information

Input and Output Segment Numbers:
IN — Segment number of input segment
OUT — Segment number of output segment

Mode Toggles:

REVERSE — Reverses point progression of curve or side of surface

SWITCH — Changes faster-running coordinate to slower-running coordinate, and vice versa

<u>**Note**</u>

The input segment must be entered first.

<u>**Example**</u>

The use of the switch function is illustrated in Fig. 58. Segment 17 is used as input into this function. The Switch toggle is selected. The slower- and faster-running coordinate directions of segment 17 are reversed in segment 25. The result of the switching operation is highlighted and displayed in the graphics window. Since no geometric data are altered, no differences between segments 17 and 25 will be evident. However, point progressions on the slower and faster-running sides may be observed manually (See Section 3.3).

3.6.4.7 Read Segment-File

This option allows a previously saved segment file (See Section 3.6.4.10) to be read and stored for further processing. The screen shown in Fig. 59 is displayed when the Read Segment-File option is selected.

<u>**Required Information**</u>

File Number:

FILE — Integer file identifier. The previously stored file must have a file name of FTxx, where xx is the file identifier + 10.

Output Segment Identifier:

SEGMENT — Segment number (See Section 3.3)

<u>**Example**</u>

In the example depicted in Fig. 59, a cubic line segment previously saved with the File option (See Section 3.6.4.10) is read. The cubic segment is stored in logical file 21 and is assigned to segment 1. The cubic curve is highlighted and displayed in the graphics area after the read operation is complete.

3.6.4.8 Read Segment List

This option allows a segment to be defined manually. Either curve or surface segments may be defined by entering the Cartesian coordinates of points on the segment. The screen depicted in Fig. 60 is displayed when the Read Segment-List option is selected.

Required Information

Dimensions:
DIMENSIONS — Integers denoting the dimensions of the segment to be defined. The second dimension will be 1 for curve segments.

Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Indices of the Segment to Be Defined:
I, J — Segment indices. J is 1 for curve segments.

Cartesian Coordinates of Points Defining Segment:
X, Y, Z — Point coordinates

Example

The example depicted in Fig. 60 illustrates the use of this option. A simple segment, consisting of four points, is manually entered and saved as segment 4. The resulting segment is highlighted and displayed in the graphics area.

3.6.4.9 Edit Segment

This option allows previously defined curve or surface segments to be edited manually. The screen depicted in Fig. 61 is displayed when the Edit Segment option is selected. Each point on the segment is listed; editing consists of selecting points and entering new coordinate locations for each selected point.

Required Information

Input/Output Segment Identifier:
SEGMENT — Segment number (See Section 3.3)

Optional Information

Toggles:

DOWN, UP — Selection of these toggles will scroll the points list.

MOD — Selection of this toggle informs the system that a point is to be modified.

Indices of Point to Be Modified:

I, J — These indices must be present in the upper table of indices.

Modified Cartesian Coordinates:

X, Y, Z — New coordinate points corresponding to indices defined next to MOD toggle

Note

All three coordinates (i.e., X, Y, and Z) must be entered if a point is to be modified, even if only one coordinate value is to be altered.

Example

In the example depicted in Fig. 61, segment 1 (the cubic curve read in the Example of Section 3.6.4.7) is edited. The coordinates of the last point in the segment are changed from (5.0, 5.0, 0.0) to (10.0, 5.0, 0.0). The modified segment is highlighted and displayed in the graphics area.

3.6.4.10 File

This option allows the user to store multiple segments into a file. The File option is usually used when all surfaces necessary for the generation of a three-dimensional grid have been developed. The screen depicted in Fig. 62 is displayed when this option is selected.

Required Information

Segments, or a Range of Segments, to be Written to a File:

SEGMENTS— Segment numbers may be entered in each box. A range of segment numbers may be specified by entering the first segment number in the first box on the left, and the last segment number in the second box. If a range is specified, the INCLUSIVE toggle must be activated.

File Output:

FILE — Output number of file (See Section 3.3)

Example

In the example depicted in Fig. 62, the segment shown in the graphics window (segment 1) is written to file number 21. A file named FT31 is produced by this operation.

3.6.5 Profiles

Surfaces of aerodynamic bodies are often described in terms of point coordinates lying in planar cross sections normal to an axis. Fuselages are usually described with planar profiles normal to the longitudinal axis, whereas wings are described relative to a spanwise axis. The cross sections, or profiles, often contain different numbers of points, and may be spaced unevenly along the body. These profile data often must be used to define a smooth, well-structured surface array, with points on the surface packed in regions where high flow gradients are anticipated.

The original batch version of the EAGLE surface generator allows input in various forms; however, it does not readily allow surfaces to be generated from profile data. The interactive version of EAGLE contains extensions to allow surfaces to be generated from sets of points obtained as profiles along a body. The menu options depicted in Fig. 63 are displayed when the Profile option is selected.

3.6.5.1 Read Profiles

This option allows the user to read in a file containing profile information. The user is prompted for the file name, which may be any valid Fortran 77 file identifier.

The profile file must be in the correct format to be utilized properly by Interactive EAGLE. The format is

```
TOTAL NUMBER OF PROFILES
NUMBER OF POINTS IN PROFILE #1
 X, Y,  Z
 .  .   .
 .  .   .
 .  .   .
```

NUMBER OF POINTS IN PROFILE #2
X, Y, Z
• • •
• • •
• • •
NUMBER OF POINTS IN PROFILE #3
etc.

Each profile is assumed to be a collection of points ordered circumferentially about some axis. In addition, each cross section is assumed to be normal to that axis. The Cartesian point data are free-format real numbers; the number of profiles and the number of points in each profile are free-format integers.

3.6.5.2 Select View Plane

This option allows the user to select the viewing plane in which the profile data will be displayed. The screen shown in Fig. 64 is displayed when the Select View Plane option is selected. The user simply selects one of the displayed toggles in order to select the desired viewing plane. If no selection is made, the default view plane is the Y-Z plane.

3.6.5.3 Edit Profile(s)

This option allows the user to interactively modify the profile data. The screen depicted in Fig. 65 is displayed when the Edit Profile(s) option is selected. Individual points can be moved, inserted, or deleted interactively with the mouse. Once corrections are made, the edited profiles can be saved in a permanent file that can be used as a starting point for future profile manipulations and/or surface generation.

3.6.5.3.1 View Profiles

This option allows the user to select any number of profiles for display. The profile viewing plane will be whatever is selected under the Select View Plane (See Section 3.6.5.2). When this option is selected, the user will be prompted for profile numbers. The user can (1) choose an individual profile number by entering a single number; (2) choose a range of profiles by entering the first profile number, a minus sign, and the last profile number (e.g., 2 − 7, for displaying profiles 2 through 7); (3) view all profiles by entering "ALL"; (4) select several profiles by entering the numbers separated by commas (e.g., "1,3,5" to view profiles 1, 3, and 5), or (5) select a combination of ranges of profiles and individual profiles (e.g., "1 − 5,7,9 − 12" to view profiles 1 through 5, profile 7, and profiles 9 through 12. In the case illustrated in Fig. 65, an X-Z view has been selected, and all profiles were selected for viewing.

3.6.5.3.2 Move

This option allows a user to interactively move a point. After selecting this option, a point can be chosen by moving the cursor over the point and pressing the middle mouse button. A window will appear above the screen that will designate the coordinates of the cursor. Moving the cursor to the desired location and pressing the middle mouse button will move the selected point to the new location. The point can be moved only along the designated viewing plane.

3.6.5.3.3 Insert

This option allows a point to be inserted between two designated consecutive points. The two points are selected by moving the cursor over the desired points and pressing the middle mouse button. After the second point is selected, a new point will appear in the graphics window. The new point can be moved to any desired location with the MOVE option (See Section 3.6.5.3.2).

3.6.5.3.4 Delete

This option allows a point to be deleted from a profile. The point to be deleted is selected by placing the cursor over the point and pressing the middle mouse button.

3.6.5.4 Save Edited Profiles

An edited set of profiles may be saved in a file by invoking this option. The user will be prompted for a file name, which may be entered as any valid Fortran 77 file identifier. The edited profiles may be read again for further processing by invoking the Read Profiles option (See Section 3.6.5.1).

3.6.5.5 Fit Around Profile(s)

This option allows a user to fit a spline curve (See Ref. 13) around a profile. The menu options depicted in Fig. 66 are displayed when this option is invoked. The fit may be done over the entire profile, or may be done piecewise in order to preserve discontinuities that may exist in the profile data. Several suboptions are available under this option.

3.6.5.5.1 View Profiles

The operation of this option is similar to the View Profiles option discussed in Section 3.6.5.3.1.

3.6.5.5.2 Select Indices

This option allows points on each profile to be selected to serve as endpoints of the spline fit. The user selects endpoints by placing the cursor on each point and pressing the middle mouse button. The endpoints are selected as pairs; however, as many pairs of endpoints may be selected as desired if piecewise fits are required.

3.6.5.5.3 Fit Profiles

Selection of this option will fit and display the splines in the regions designated by the point selection process (See Section 3.6.5.5.2). If no endpoints are selected, the fit will be done over the entire profile.

The spline curves will be subsequently used to distribute points; therefore, at this stage the user can examine the curve fits to see if the integrity of the surface description has been maintained. If the points must be fit again, the point selection process must be repeated with new endpoints.

In the case of piecewise fitting of the profiles, care should be taken that each resulting curve adjoins (but does not overlap) another curve. Regardless of the order in which the endpoints are specified, the resulting curves are ordered (and therefore identifiable) with respect to their smallest index, e.g., a curve between points 10 and 24 will precede a curve between points 24 and 35. In addition, the "first end" of a curve is the end with the smaller index; the "last end" has the higher index. In the example depicted in Fig. 66, all profiles have been fitted.

3.6.5.5.4 Redraw Screen

This option simply redraws the graphics area with the latest information that is available.

3.6.5.6 Pack Around Profile(s)

Once the profiles have been represented with splines, points can be distributed along the splines according to specified packing criteria. The menu options depicted in Fig. 67 are displayed when this option is selected. The suboptions available under this option are discussed in the following sections.

3.6.5.6.1 View Profiles

The operation of this option is similar to the View Profiles option discussed in Section 3.6.5.3.1.

3.6.5.6.2 Pick Packing and Points

This option allows the user to specify the type of packing, spacings, and the number of points to be applied on selected profiles. The user is prompted for the following information:

Profile #(s) — The profile(s) for which the packing and spacing information pertains. The user can

1. choose an individual profile number by entering a single number,
2. choose a range of profiles by entering the first profile number, a minus sign, and the last profile number (e.g., 2 – 7, for displaying profiles 2 through 7),
3. select all profiles by entering "ALL,"
4. select several profiles by entering the numbers separated by commas (e.g., "1,3,5" to select profiles 1, 3, and 5), or
5. select a combination of ranges of profiles and individual profiles (e.g., "1 – 5,7,9 – 12" to select profiles 1 through 5, profile 7, and profiles 9 through 12.

Each profile selected must be fitted with the same number of spline sections (See Section 3.6.5.5.3).

Packing(s) — Three types of packing are supported. For each curve defining a profile, points may be packed at both ends, at the "first end," or at the "last end." These packings are specified by entering a 1, 2, or 3, respectively. If more than one curve is used to define a profile, then the appropriate number of packing specifications must be entered. For instance, if three curves are used to define a profile, the packing could be entered as "1,3,2", e.g., pack the first curve at both ends, the second curve at the "last end," and the third curve at the "first end."

Spacing(s) — The relative spacings (See Section 3.2) at the endpoints of each curve defining a profile must be specified. Two spacings must be entered for a type 1 packing; one spacing must be entered for types 2 and 3. For the packing example above, four spacings must be entered. Entering spacings of ".01,.02,.05,.07" would apply spacings of 0.01 and 0.02 to the first and last ends of the first curve, respectively. A relative spacing of 0.05 would be applied to the "last end" of the second curve; a spacing of 0.07 would be applied to the "first end" of the third curve.

Points — The number of points along each curve fit must be specified. These points will be distributed along each curve according to the type of packing and the spacings specified. In the present example, entering "20,10,15" would distribute 20 points along the first curve, 10 along the second, and 15 along the third. Since two endpoints of the three curves will be coincident, a curve containing 43 points will result.

Note: This screen was designed so the user does not have to use the mouse until all desired packing information has been entered. The RETURN key moves the cursor into the required columns. Entering RETURN in the "Profile #(s)" column will terminate entry.

In the example depicted in Fig. 67, points have been packed at both endpoints of the curves resulting from the fitting process depicted in Fig. 66.

3.6.5.6.3 View Packed Profiles

When packing has been performed, the result may be viewed by invoking this option. The system will prompt the user for the profiles to be displayed. The previously described conventions for designating profiles (See Section 3.6.5.6.2) may be used.

3.6.5.6.4 Redraw Screen

This option simply redraws the graphics area with the latest information that is available.

3.6.5.7 Fit Down Profile(s)

The operation of this option is identical to that of the "Fit Around Profile(s)" option described previously, except that the functions are being performed in the longitudinal, rather than the circumferential, direction. Refer to Section 3.6.5.5 for information on the operation of this option. The screen depicted in Fig. 68 illustrates the use of the longitudinal fitting function. In the example depicted, all profiles have been fitted longitudinally.

3.6.5.8 Pack Down Profile(s)

The operation of this option is identical to that of the "Pack Around Profile(s)" option described previously, except that the functions are being performed in the longitudinal, rather than the circumferential, direction. Refer to Section 3.6.5.6 for information on the operation of this option. The screen depicted in Fig. 69 illustrates the results of the longitudinal packing procedure. In the example depicted, the fuselage points have been packed near the nose.

3.6.5.9 Connect Profile(s)

This option provides the interface between the profile processor and the rest of the Interactive EAGLE surface generator. The suboptions available under this option are discussed in the following sections.

3.6.5.9.1 Connect Profiles

The surface points generated using the previously described fitting and packing operations may be connected to yield a two-dimensional surface mesh. When this option is invoked, the user is prompted for a range of "J" and "K" indices, where "J" and "K" are assumed to be in the longitudinal and circumferential directions, respectively. These ranges must be specified as pairs of integers, separated by commas. For instance, if the fitting and packing process resulted in 30 longitudinal points and 25 circumferential points, the respective ranges would be specified as "1,30" and "1,25." The screen depicted in Fig. 70 illustrates a surface segment that has resulted from the previously described fitting and packing functions.

3.6.5.9.2 Store Segment

The surface segment generated under the Connect Profiles option (See previous section, Section 3.6.5.9.1) is now ready for use as an EAGLE segment. The Store Segment option allows a user to store the surface description obtained from the fitting and packing process as an EAGLE segment. The user is prompted for ranges of indices corresponding to the points that are to be saved for the surface description (See Connect Profiles, Section 3.6.5.9.1), and a segment number (See Section 2.4). If an output restart file has not been specified, the user will be asked to supply a restart file name; the segment description will be saved as part of the restart file (See Section 3.5.1).

3.6.5.9.3 Restart Save/No Save

This is a toggle that specifies whether the surface segment is to be saved with the restart file (See Section 3.5.1). Normally, the segment will be saved unless otherwise specified.

4.0 GRID GENERATOR — GENERAL OPERATION

4.1 GRID GENERATION SYSTEM

The Interactive EAGLE grid code is based on general three-dimensional algebraic and elliptic grid generation techniques (Ref. 4). The algebraic three-dimensional generation system is based on transfinite interpolation (using either Lagrange or Hermite interpolation), and is generally used to generate an initial grid to start the iterative solution of the elliptic generation system. However, the code may be run as a strictly algebraic generation system if desired.

The elliptic generation system allows automatic evaluation of control functions either directly from the initial smoothed algebraic grid, or by interpolation from the boundary-point distributions. In the former case, the smoothing is done only in the two directions other than that of the control function. This allows the relative spacing of the algebraic grid to be retained, but results in a smoother grid than that produced by the algebraic procedure. In the latter case, the arc length and curvature contributions to the control functions are evaluated and interpolated separately into the field from the appropriate boundaries. The control function at each point in the field is then formed by combining the interpolated values. This procedure allows very general regions with widely varying boundary curvature to be treated successfully.

The control functions can also be determined automatically to provide orthogonality at boundaries with specified normal spacing. The amount of orthogonality in the field can also be controlled on specified grid surfaces within the field, allowing grid surfaces in the field to be kept fixed while retaining continuity of slope of the grid lines crossing the surface. This has proven to be very useful in controlling the skewness of grid lines in troublesome areas. Alternatively, boundary orthogonality can be achieved through Neumann boundary conditions that allow the boundary points to move over a surface spline. Provision is also made for extrapolated zero-curvature boundary conditions and for mirror-image reflective boundary conditions on symmetry planes.

The grid code allows any number of blocks to be used to fill an arbitrary three-dimensional region. Any block can be linked to any other block (or to itself) with partial or complete continuity across the block interfaces. In the case of complete continuity, the interface is a branch cut, and the code establishes a correspondence across the interface using a surrounding layer of points outside the blocks. This allows points on the interface to be treated just as all other points so that there is no loss of continuity. The physical location of the interface is thus unspecified, and is determined by the code. Although written for three-dimensional grid generation, the code can operate in a two-dimensional mode on either a plane or curved surface. In the case of a curved surface, the surface is defined with splines, and the surface grid is generated in terms of parametric surface coordinates.

4.2 OPERATION OF THE GRID GENERATOR

The operation of the grid generator will be illustrated with the example of a two-dimensional airfoil mesh. A two-dimensional example is chosen for clarity and conciseness; however, the generation of three-dimensional meshes will follow the same general procedure.

4.2.1 Screen Organization

All screens in the grid generator are arranged in essentially the same manner. The screen depicted in Fig. 71 illustrates the general layout. A window in the upper right corner contains options that may be selected at any time during the operation of the grid generator. The large upper window is the "physical" window, and is used to display a physical depiction of the segments that are to be used as boundaries for the grid generation algorithms. The large lower window contains graphical depictions of the functions to be performed, and is referred to as the "function window." The function window corresponds roughly to the "forms" that are used in the surface generator (See Section 2.2). The rightmost window (the "points" window) contains information that allows the user to (1) select boundaries, and (2) impose boundary conditions and various types of interpolations on the selected boundaries. The window to the left of the points window is referred to as the "segments" window, and contains information that allows the user to identify and select different segments for processing. The leftmost window, the "boundary condition" window, is a table that identifies which boundary conditions have been associated with the various segments. Two other auxiliary windows, the "text" and "output" windows, allow the user to enter data and view output from the grid generator, respectively.

4.2.2 Selection of Menu Options

All menu options are selected by placing the cursor on the option and pressing the left mouse button. When the cursor is placed on an option, the option will be highlighted.

4.2.3 Control of the Graphics Display

The graphical depiction in the physical window is controlled in a manner similar to that used in PLOT3D (Ref. 12). The cursor must be placed in the physical window in order to alter the view of the depicted segments. The left mouse button controls rotation, the middle mouse button rotation (when moved left and right) and zoom (when moved up and down), and the right mouse button controls translation.

4.2.4 Miscellaneous Controls

The restart file generated during an interactive session may be viewed at any time by placing the cursor in the output window and pressing the right mouse button. The function window will then display the text of the restart file, as well as any error messages and output from the grid generator. The Escape key must be pressed to continue interactive operation.

In the event it is necessary to quit an interactive operation (e.g., in the event an error has been made), the cursor can be returned to the menu at the upper right corner by pressing the Escape key. The interactive session may be continued by selecting any desired operation from the menu.

4.3 MAIN MENU

When the grid generator is invoked, the user will be asked if Direct, Borrow, or No Graphics mode is desired. Direct and Borrow modes are described for the surface generator in Section 3.4. The grid generator can be executed in batch mode by selecting the No Graphics mode. To use the No Graphics mode, the restart file must first be edited to insert an "N" as the first record. The grid generation code may then be run in the No Graphics mode using redirected input from the restart file.

If Direct or Borrow mode is selected, the main menu will appear. The options available under the main menu are those depicted in Fig. 71, and are described in the following sections.

4.3.1 Block

The purpose of this option is to designate (1) how many blocks will comprise the final grid configuration, and (2) the dimensions of each block. The screen depicted in Fig. 72 is presented when the Block option is selected.

4.3.1.1 Open

This option must be selected for each block in which a grid is to be generated. Because of memory restrictions, the present interactive implementation is restricted to two blocks; however, an arbitrary number of blocks could in principle be designated. For the present example of a two-dimensional airfoil, this option is selected once.

4.3.1.2 Size

This option designates the size (i.e. the dimensions) of a selected block. In the event more than one block has been opened, a block must first be selected before this option is invoked (See Section 4.3.1.3). If only one block has been designated, this option may be invoked without explicitly selecting the block.

When this option is invoked, a diagram will appear in the lower large window, as depicted in Fig. 72. This diagram represents the computational volume in which a grid is to be generated. The dimensions are entered by placing the cursor in the left box under "Dimensions" and pressing the left mouse button. At this point the dimension corresponding to the first coordinate direction may be entered. When the RETURN key is pressed, the cursor will jump to the second box. At this point the dimension corresponding to the second coordinate direction may be entered. The process is repeated for the third coordinate direction. The cursor may then be moved to the "Done" box; pressing the left mouse button completes the designation of the block dimensions.

In the example depicted in Fig. 72, the block is designated as having dimensions of 91, 41, and 1, respectively. The designation of the third dimension indicates the resulting mesh will be two-dimensional.

4.3.1.3 Select Block

The mesh blocking capabilities of EAGLE are incorporated into the interactive grid generator. Since mesh blocks are treated as separate meshes, each block that has been opened (if there is more than one) must be selected prior to subsequent processing. When this option is invoked, a menu containing previously opened block numbers will appear in the Points window. A block is selected by placing the cursor on the appropriate block number and pressing the left mouse button. All subsequent operations are then applied to the selected block until another block is selected.

4.3.2 Read

Selection of this option allows the user to read files that have been previously generated with the surface generator and stored using the File option (See Section 3.6.4.10). The screen depicted in Fig. 73 will appear when this option is selected from the main menu.

4.3.2.1 Segments

When this option is selected, the user will be prompted for a file number and a file name. The prompts will appear in the text window; the file information is also entered in the text window. The file number given is arbitrary; the file name corresponds to that given under the File option of the surface generator (See Section 3.6.4.10).

In the example depicted in Fig. 73, four segments that have been previously generated by the surface generator have been read. The boundaries consist of four curve segments that will serve as the boundaries of a two-dimensional airfoil mesh.

In the case of a three-dimensional mesh, surfaces instead of curve segments must be used as boundaries. However, the surfaces may be defined only by their boundary curves; the grid generator will interpolate surface descriptions if necessary. For the purposes of the two-dimensional airfoil example, the curve boundaries will be referred to as "surfaces" for the sake of consistency.

4.3.3 Place

The surface generator produces surface descriptions that are not associated with any particular face of the computational volume. In addition, the coordinate progressions of the surfaces may not be in the directions required for proper mapping to the computational volume. The Place option performs the mapping necessary to associate the surface segments generated by the surface generator with faces on the computational volume. In addition, the Place option allows the user to designate boundary conditions on the different faces, and to designate different types of interpolations to be used on faces that, in physical space, are defined only by boundary curves. The screen depicted in Fig. 74 appears when the Place option is selected from the main menu.

4.3.3.1 Segments

The purpose of this option is to associate segments in the physical window with the appropriate sides of the computational volume. The function window depicted in Fig. 74 appears when the Segments option is selected. The function window contains depictions of the six sides of a computational volume. For the present two-dimensional example, only one side of the computational volume (i.e., either the L = 1 or the L = LMAX side) must be associated with the four segments that have been previously read.

The segment numbers and dimensions of the segments that have been previously read (See Section 4.3.2) are displayed in the segment window. Placing the cursor on a row in the segment window will highlight the corresponding segment in the physical window. In this manner, segments in the physical window can be easily scanned and identified. Highlighted segments will have a circle displayed at the first point on the segment. In the example of Fig. 74, a circle is displayed at the first point on segment 12.

A face on the computational volume may be selected for processing by placing the cursor on one of the faces and pressing the left mouse button. When this is accomplished, the face will be surrounded by a box (See Fig. 74). The computational surface may be made clearer by pressing the Z key, which will enlarge the face to fill the entire function window, as depicted in Fig. 75. Initially, a circle will appear at the lower left corner of the computational surface (e.g., at J = 1 and K = 1).

The circles on a particular segment and the computational surface must be associated with each other. A segment to be associated with the displayed computational face may be selected by placing the cursor on the appropriate segment number in the segment window and pressing the left mouse button. The circle on the computational surface may be moved via the arrow keys to correspond to the desired computational coordinate of the circle displayed on the selected physical segment. In the present example, the circle on segment 12 is to be associated with the J = 1, K = KMAX computational coordinate. This is accomplished by moving the circle on the computational surface to J = 1, K = KMAX and pressing the RETURN key. The circle placement is depicted in Fig. 75.

The next step is to associate a direction on the computational surface with a direction on the physical segment. In the present example, the direction in computational space will be starting at J = 1, K = KMAX and progressing to J = JMAX. This direction may be selected by pressing the right arrow key. (The "down" arrow key could be used if the desired direction was from K = KMAX to K = 1.) A green bar will appear along the corresponding side of the computational surface; pressing RETURN will associate segment 12 with the entire K = KMAX surface from J = 1 to J = JMAX. At this point the green bar will be replaced with a black bar as depicted in Fig. 75. The corresponding physical segment will be displayed as a dashed line; in this manner it can be seen which computational surfaces have been associated with corresponding physical segments.

The process described is repeated for the other three segments. The Segment option must be selected for each segment that is to be mapped into the computational space. At the completion of the mapping process, the selected face will be completely surrounded by a black border.

4.3.3.2 Auto

At some point in the mapping process, there is enough information to automatically associate selected segments with computational surfaces and coordinate directions. If a segment is selected, and the Auto option is invoked, the system will attempt to perform the mapping for the selected segment automatically. This option should be used with caution, however, since it will fail in regions where surfaces are degenerate, e.g., at polar axes or cuts.

4.3.3.3 Type

After the segment mapping process (See Section 4.3.3.1) has been completed, boundary conditions must be associated with each computational face. Selection of the Type option will result in the display of the screen depicted in Fig. 76. On the right is a window (the Points window) denoting all endpoints of the segments that have been associated with the L = 1 computational surface. Placing the cursor over a row in the Points window will highlight the associated point in both the function and physical windows.

A single boundary condition (e.g., "fixed" points), may be associated with all segments by placing the cursor on "All" in the Points window and pressing the left mouse button.

A menu with the available boundary conditions will appear in the points window (See Fig. 77); selecting from the menu will associate the chosen boundary condition with all segments in the physical window that have been associated with the selected computational surface.

Applying a single boundary condition to every segment may result in a suitable mesh for many applications. However, it is often desirable to apply different boundary conditions to different segments. In the present example, for instance, it would be advantageous to apply orthogonal boundary conditions to the K = 1 surface. The K = 1 surface falls between points 3 and 4. This could be confirmed by scanning through the points list with the cursor and watching the points displayed in the physical and function windows. Placing the cursor on point 3, pressing the left mouse button, and repeating for point 4 will select the associated segment. The boundary condition menu will appear (See Fig. 77), at which point a desired boundary condition may be selected. This process may then be repeated for the other segments.

When a boundary condition is associated with a segment, a colored bar will appear in the computational window along the line corresponding to the associated physical segment. The type of boundary condition will appear in the boundary condition window. In the present example depicted in Fig. 77, the words "FIX" and "ORTHOG" have appeared in the top of the boundary condition window. The lettering will be in the color of the bars denoting the segments upon which boundary conditions have been imposed. In this manner different boundary conditions can be associated with different colors, and the user can conveniently review which boundary conditions have been assigned to which segments.

After all segments associated with a given computational surface have been assigned boundary conditions, the boundaries of the graphical depiction of the computational surface will be bordered by colored bars. At this point the user will know that all boundary condition information needed for the surface has been provided. For the two-dimensional example, only the L = 1 surface needs to be provided boundary condition information; in a three-dimensional case, the described process for imposing boundary conditions must be repeated for the remaining five computational boundaries.

4.3.3.4 Interpolation

The Interpolation option allows the selection of several variations of transfinite interpolation to be applied to the faces or interior of a computational volume. The screen depicted in Fig. 78 is displayed when the Interpolation option is selected. Different variations may be specified for individual segments (i.e. different faces) or the same variation may be selected for all segments. The specification of individual segments is accomplished in the manner described for imposing boundary conditions (See Section 4.3.3.3).

Four types of interpolations may be specified: Lagrange, Hermite, Hermite1, and Hermite2. These interpolation types relate to the forms of the blending functions used. Different types may be selected for each computational direction by placing the cursor at the desired row (selecting the type) and column (selecting the computational direction) and pressing the left mouse button. (The numbers 1, 2, and 3 correspond to the J, K, and L computational coordinates, respectively.) Similarly, arc length and linear forms of the blending functions may be selected for each computational direction by (1) selecting either ARC or LINEAR and (2) selecting a computational direction from the "DIRECTIONS" box. Finally, the interpolation may be designated as being from the faces, the edges, or the corners of the computational region, by selecting "FACES," "EDGES," or "CORNERS," respectively. If "ALL" is selected, the interpolation will be performed from all defined segments.

For the two-dimensional case depicted in Fig. 77, four segments have been designated as edges of the computational region. Lagrange interpolation is selected in both computational directions, and blending functions based on arc length are selected. In this case, whether "ALL," "FACES," or "EDGES" is selected will not affect the interpolation.

Once the desired interpolation has been selected, the RETURN key is pressed. The grid generator will then proceed to perform the interpolation on the selected face of the computational volume. In the present example, the screen depicted in Fig. 79 is displayed when the interpolation is completed. The physical window contains the algebraic grid generated through the interpolation. The function window depicts the associated computational face completely blacked out, indicating that the interpolation has been performed. Colored bars associated with the selected boundary conditions remain, as the boundary conditions apply only to the generation of elliptic grids. If an elliptic grid is generated, the displayed algebraic grid is used as an initial grid for the elliptic solver.

4.3.4 Utility

The screen depicted in Fig. 80 is presented when the Utility option is selected. The Utility screen allows the selection of several options for grid generation.

If only an algebraic grid is desired, the "ALGEBRAIC GRID" box may be selected. The Utility screen may then be exited without entering further information. Selecting the Generate option from the main menu will then result in the generation of an algebraic grid (See Section 4.3.6), using the interpolation information provided from the Interpolation option (See Section 4.3.3.4).

If an elliptic grid is desired, additional information must be entered. The number of iterations may be entered in the ITERATIONS box. A convergence tolerance may be entered in the TOLERANCE box; if the residual calculated during the solution of the elliptic equations falls below TOLERANCE, the solution will terminate before the maximum number of iterations have been completed. If TOLERANCE is omitted, the number of iterations specified will be completed. An acceleration parameter may be entered in the designated box. This parameter is a real number between 0 and 2. Normally, it should be set to about 1.8; however, it must be decreased if the grid is tightly packed. If ACCELERATION PARAMETER is omitted, an optimum acceleration parameter is calculated and used.

Control functions may be determined by the modes depicted in the center column of boxes. Selection of the RADIUS mode results in control functions being evaluated from the boundary point arc length distributions on the sides of the block. Selection of the INITIAL mode results in control functions being evaluated from interpolations from the initial algebraic grid. The interpolations are smoothed before being used in the elliptic equations. This method of evaluating control functions returns a grid that will retain the general grid line distribution of the initial algebraic grid; however, the resulting grid will exhibit more smoothness and less skewness than the algebraic grid. Use of the INITIAL option is the preferred mode of operation.

Selection of the NONE box results in the control functions being set to zero. The result is a Laplace grid. Selection of the AVERAGE CUTS box will average all control functions across a cut.

The forms of derivatives used in the finite-difference approximations of the elliptic equations may be selected from the right column. First and cross derivatives may be designated as VARIABLE, ONE-SIDE, and CENTRAL. Selection of the VARIABLE box will result in the use of central differences for control functions that do not exceed a value of 2.0, with weighted averages between central and one-sided differences otherwise. The ONE-SIDE option specifies the evaluation of one-sided differences. For first derivatives, the direction of the differencing is dependent on the sign of the control function. For the cross derivatives, the direction of the differencing depends on the sign of the off-diagonal metric elements. The CENTRAL option specifies the use of central differences for all first and cross derivatives.

4.3.5 Output

Upon selection of this option, the user will be prompted for a file number and name; the output of the grid generation process will then be stored in the designated file. The file number is arbitrary, but must be different from those specified under the Read (See Section 4.3.2) option. The file name must be under eight characters long, including extensions.

4.3.6 Generate

Selection of this option begins the elliptic grid generation process. The options previously selected pertaining to the surface mappings, boundary conditions, types of interpolations, and type of grid generation (i.e. elliptic or algebraic) are applied at this point. When this option is invoked, all windows and menus disappear, and the program continues in essentially a batch processing mode. The normal EAGLE output will appear on the screen, and can be reviewed for indications of error or grid abnormalities. If the grid generation is successful, the resulting grid points are written to the file specified (See Section 4.3.5). Presently, the interactive grid generator output is in PLOT3D format (Ref. 12) . As a result, the grid may be conveniently viewed with PLOT3D, but must be converted via a utility program to be put in a suitable format for a given flow solver.

The grid generation process initiated by the selection of this option may be lengthy, especially in the case of generating elliptic grids. However, the restart file generated up to this point may, if desired, be used on a mainframe as input for the batch EAGLE grid generator. In the present example, the grid depicted in Fig. 81 was generated from the inputs provided through the interactive session.

4.3.7 Restart

This option allows a previously generated restart file to be processed. The operation of this option is similar to the restart processing options in the surface generator (See Section 3.5.1). However, in the grid generator, the restart file is always stored in a file named "gridgen.jou." This file must be renamed in order to be used by the grid generator.

The restart file may be examined at any time during an interactive session by placing the cursor in the output window (the bottom window) and pressing the right mouse button. The options available for processing restart files are as follows:

OPEN

Upon the selection of this option, the user will be prompted for the name of the restart file to be processed. This option must be invoked before any of the other restart file options.

CLOSE

This option closes the current restart file. Another restart file could subsequently be opened and processed.

ACCEPT

This option processes one line of the restart file. This "stepping" mode may be used if a subsequent command in the restart file is to be altered or skipped.

SKIP

Selection of this option will result in the skipping of the next instruction in the restart file. This option is used if the user wishes to modify the instruction through the graphical interfaces.

ACCEPT THE REST

Selection of this option will result in the processing of all remaining instructions in the restart file.

5.0 EXAMPLE OF INTERACTIVE SURFACE AND GRID GENERATION

Figures 82 through 89 depict selected screens presented to a user developing a three-dimensional grid from an initial set of cross-sectional surface data. The starting point for this example is the surface generated from the examples of Section 3.6.5. The result of the fitting and point distribution process is depicted in Fig. 82, which shows the surface of the aerodynamic configuration derived from the initial cross sections.

It is advantageous to develop tables of geometric objects and attributes that can later be referenced indirectly from the forms that generate various geometric constructs. The advantages of indirect referencing are described in Section 2.4. Figure 83 depicts three such tables. Figure 83a is a set of points and associated coordinate locations in three-dimensional space; Fig. 83b is a list of real numbers that correspond to various mesh spacings; and Fig. 83c is a list of integers that correspond to the number of points along a line or along a boundary of a surface or grid. The table entries will be referenced later in the interactive session by their positions (i.e., their indices) in the tables.

The graphical definition of a polar axis is depicted in Fig. 84. The interface for defining straight lines has been invoked. The user must identify the endpoints and the number of points to be distributed on the line, along with spacing information. In addition, an identifying integer to be associated with the new line element must be supplied. The information supplied by the user is highlighted for clarity. In this example, the endpoints and spacings are referenced indirectly by specifying the labels (i.e. table entry indices) of the previously defined points and spacings. When graphical objects are referred by labels, relevant information (e.g., coordinate values) is displayed in appropriate locations in the interface.

When all necessary information is supplied, the user informs the system through the mouse interface that input has been completed. EAGLE then performs the line generation, and displays the result in the graphics window in the lower left corner of the interface. After the polar axis is defined, the line generation interface is also used to describe a straight line (designated as segment 8 in Fig. 85), which defines the intersection of the symmetry plane and the downstream boundary.

The definition of the intersection between the outer boundary and the symmetry plane is illustrated in Fig. 85. The interface for defining general cubic space curves has been invoked. To define a cubic curve, the user must specify the tangent vector at each endpoint, in addition to the endpoints themselves. In this example, a cubic curve has been constructed between the endpoints of the previously defined vertical straight line and the polar axis. Again, indirect referencing is used to identify the desired endpoints and number of points along the cubic segment. Since no spacing along the curve is explicitly specified, a default spacing is invoked; in this case, the desired number of points is simply distributed evenly along the cubic segment.

At this juncture, sufficient information exists to generate a surface on the symmetry plane. Several methods of generating surfaces are available. The screen used to generate transfinite interpolations has been invoked for this example, as depicted in Fig. 86. The user simply defines which curves are to be used as boundaries of the surface by supplying the appropriate identifiers. (Segment 10 is the curve defining the intersection of the symmetry plane and the lower fuselage surface.) In addition, the identifier of the new surface must be supplied by the user. The result is displayed in the graphical window. The generation of the upper symmetry plane surface is accomplished in a similar manner.

The generation of the outer boundary is illustrated in Fig. 87. The surface of the outer boundary is generated by rotating the previously generated cubic curve (i.e. segment 9) about the polar axis. The final step in the surface generation process is to generate the downstream boundary. This is accomplished in the same manner in which the symmetry planes were generated.

At this point, six surfaces (the body surface, the polar axis, two symmetry planes, the downstream boundary, and the outer boundary) have been defined. (The polar axis is a degenerate surface.) However, no correspondence between physical surfaces and the computational domain has yet been established. The physical surfaces must be mapped into the desired computational domain in order for a three-dimensional grid to be generated. The mapping process is depicted in Fig. 88. The top window depicts the surface elements generated previously. The bottom window depicts the six sides of the computational domain. The computational domain is defined as extending from $J = 1$ to JMAX, $K = 1$ to KMAX, and $L = 1$ to LMAX, where JMAX, KMAX, and LMAX are the maximum dimensions in the computational coordinates. The surfaces in the top window can be identified and selected by placing the mouse in one of the boxes of the segment list. (The segment list is highlighted for clarity.) The mapping process for a single surface is accomplished by (1) selecting the surface, (2) selecting a particular direction on the surface, and (3) selecting a corresponding side and direction of the computational domain into which the selected surface is to be mapped. This process is then repeated for the other five surfaces. In this example, the body surface (segment 1), is mapped to the L = 1 computational surface. The outer boundary (segment 12) is mapped to the L = LMAX surface. The lower and upper symmetry planes (segments 11 and 18, respectively) are mapped to the K = 1 and K = KMAX surfaces, respectively. Finally, the polar axis (segment 5) is mapped to the J = 1 surface, and the downstream boundary (segment 21) is mapped to the J = JMAX surface.

Once the mapping to the computational domain is completed, a three-dimensional grid can be generated. An algebraic grid can be generated with no further information. If a grid based on the elliptic solver is desired, boundary conditions must be defined. Boundary points may be interpolated (i.e. "floated" on the surface), fixed, or positioned to impose orthogonality at the surface. An algebraic grid is used as an initial mesh; the elliptic solver will then impose the required boundary conditions and control functions during the solution process. The final result is depicted in Fig. 89, which shows an algebraic grid resulting from the surface generation and mapping processes.

6.0 CONCLUSION

Interactive EAGLE, an interactive, graphically oriented surface mesh and three-dimensional grid generation system based on the EAGLE grid generation codes, has been implemented in a graphics workstation environment. Interactive EAGLE currently runs on Apollo workstations, and is available in a version that runs on workstations using the PHIGS standard graphics library. Interactive EAGLE incorporates nearly all of the features of the EAGLE codes and has been extended to accommodate profile input. During operation, the description of the surface and grid generation process is preserved. The preserved description can be modified and re-executed by Interactive EAGLE if the user wishes to alter any portion of the geometry without interactively repeating the entire input stream. In addition, the preserved process can be used as input by batch versions of EAGLE. Under test conditions, Interactive EAGLE has typically increased the throughput of surface and mesh generation by a factor of five over existing batch-oriented methods.

Interactive EAGLE may be obtained by contacting AEDC/DOCS, Arnold Engineering Development Center, Arnold AFB, TN, 37389.

REFERENCES

1. Thompson, J. F. "A Composite Grid Generation Code for General 3D Regions-The EAGLE Code." *AIAA Journal*, Vol. 26, No. 3, 1988, p. 271.

2. Lijewski, L. E. et al. "Program EAGLE User's Manual. Volume I-Introduction and Grid Applications." AFATL-TR-88-117, 1988.

3. Thompson, J. F. and Gatlin, B. "Program EAGLE User's Manual. Volume II—Surface Generation Code." AFATL-TR-88-117, 1988.

4. Thompson, J. F. and Gatlin, B. "Program EAGLE User's Manual. Volume III—Grid Generation Code." AFATL-TR-88-117, 1988.

5. Martinez, A. "Standardization of Program EAGLE—Numerical Grid Generation." MS Thesis, Mississippi State University, 1988.

6. Thompson, J. F., Lijewski, L. E., and Gatlin, B. "Efficient Application Techniques of the EAGLE Grid Code to Complex Missile Configurations." AIAA-89-0361, 27th Aerospace Sciences Meeting, Reno, Nevada, 1989.

7. Thompson, J. F. "A General Three-Dimensional Elliptic Grid Generation System on a Composite Block Structure." Computer Methods in Applied Mechanics and Engineering, Vol. 64, 1987, p. 377.

8. Thompson, J. F. "A Survey of Composite Grid Generation for General Three-Dimensional Regions." *AIAA Progress in Astronautics and Aeronautics*, Vol. 102, Numerical Methods for Engine-Airframe Integration, S. N. B. Murthy and S. G. Paynter, (Editors), 1986, p. 52.

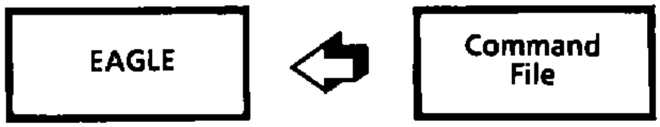
9. Vinokur, Marcel. "One-Dimensional Stretching Functions for Finite-Difference Calculations." *Journal of Computational Physics*, Vol. 107, 1985, p. 241.

10. Thompson, J. F. and Mastin, C. W. "Order of Difference Expressions in Curvilinear Coordinate Systems." *Journal of Fluids Engineering*, Vol. 107, 1985, p. 241.

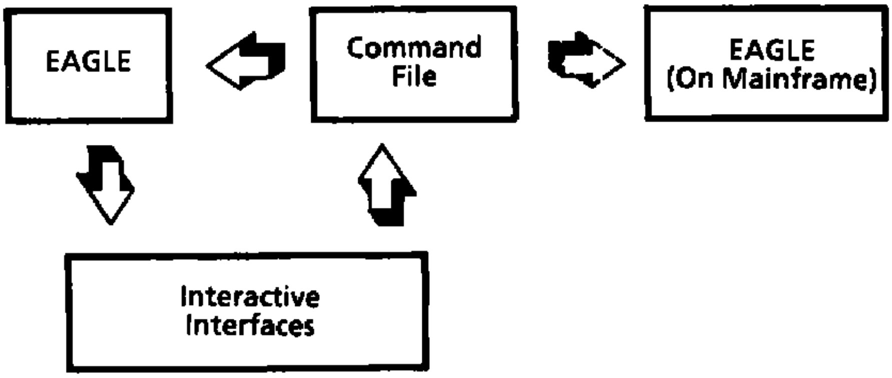
11. Jones, G. A., Thompson, J. F., and Warsi, Z. U. A. "Surface Grid Generation for Composite Block Grids." In *Numerical Grid Generation in Computational Fluid Dynamics*, Sengupta et al. (Editors), Pineridge Press, 1988, p. 1967.

12. Buning, P. and Steger, J. L. "Graphics and Flow Visualization in Computational Fluid Dynamics." AIAA Paper 85-1507, 1985.

13. Akima, H. "A Method of Smooth Curve Fitting." ESSA Technical Report, ERL 101-ITS 73, 1969.



a. Batch EAGLE



b. Interactive EAGLE
Figure 1. Data flow of batch and interactive EAGLE.

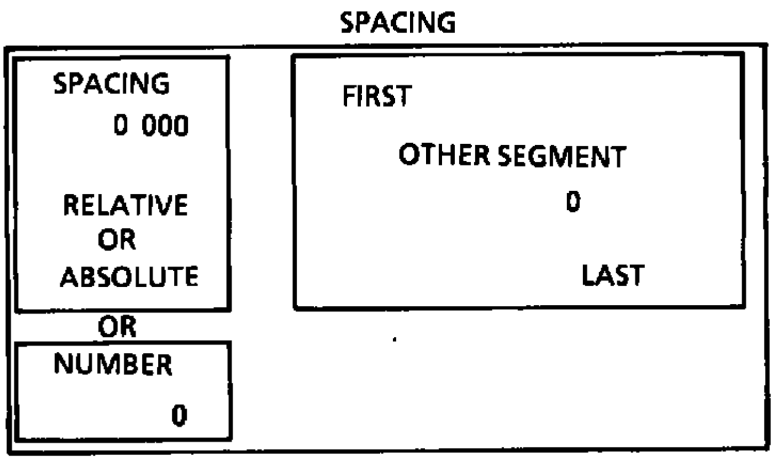


Figure 2. Spacing entry box.

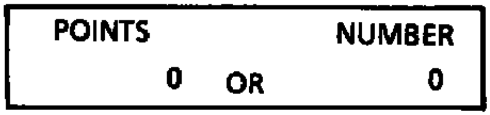


Figure 3. Points entry box.

Figure 4. Segment entry box.

Figure 5. Master menu.

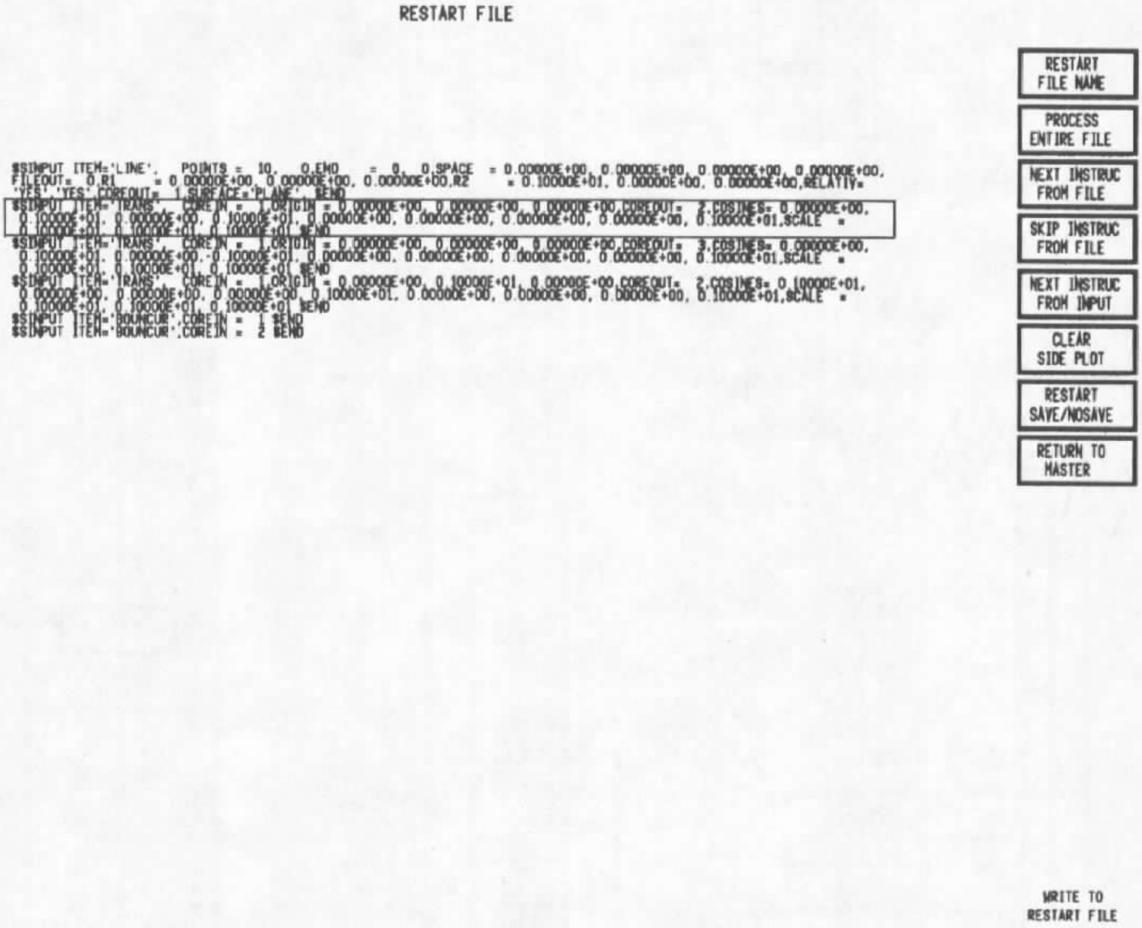


Figure 6. RESTART FILE form.

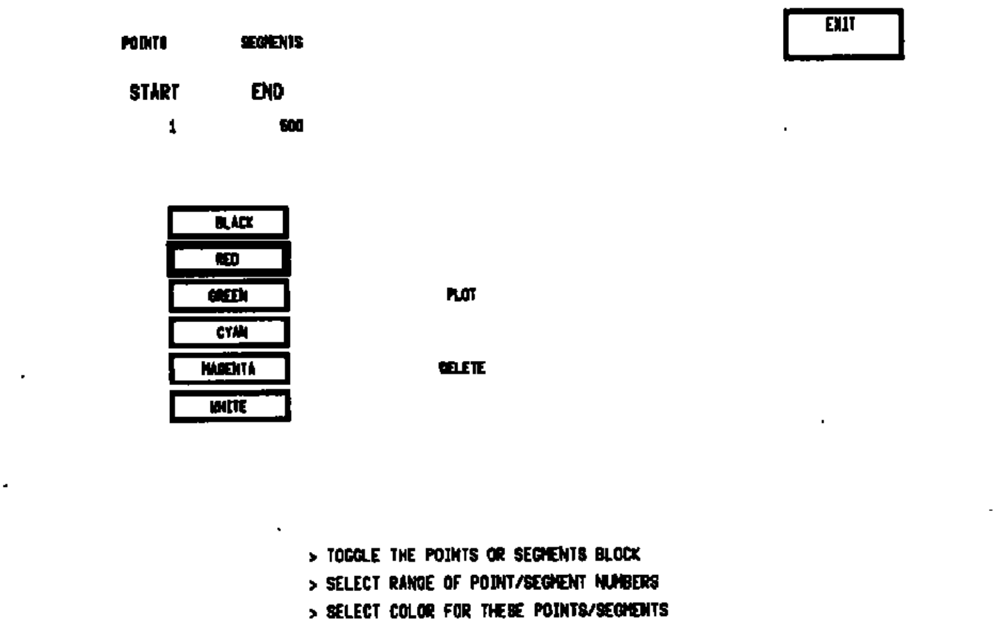


Figure 7. PLOT SEGMENTS form.

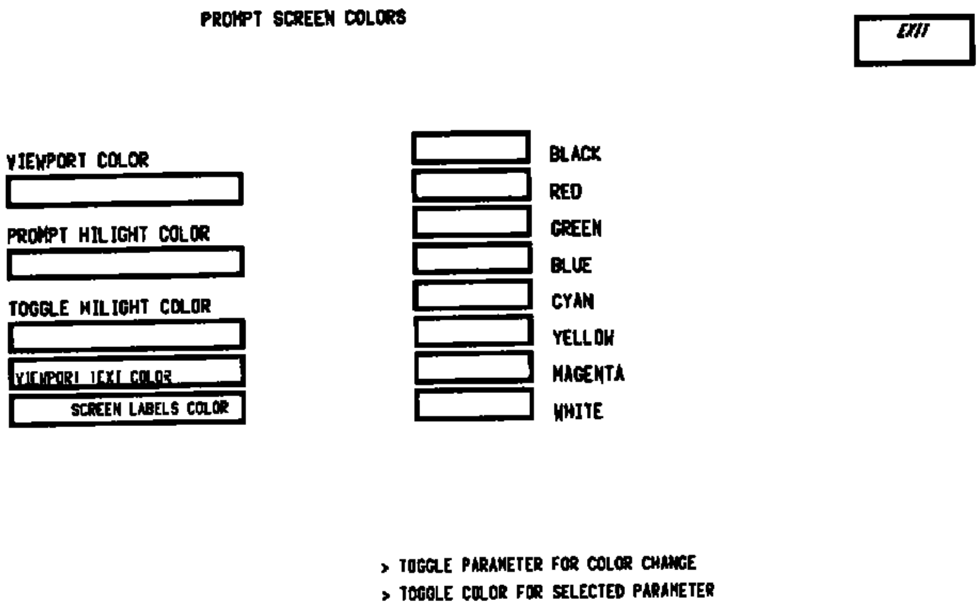


Figure 8. SET SCREEN COLORS form.

Figure 9. SCREENS menu.

Figure 10. SETTINGS menu.

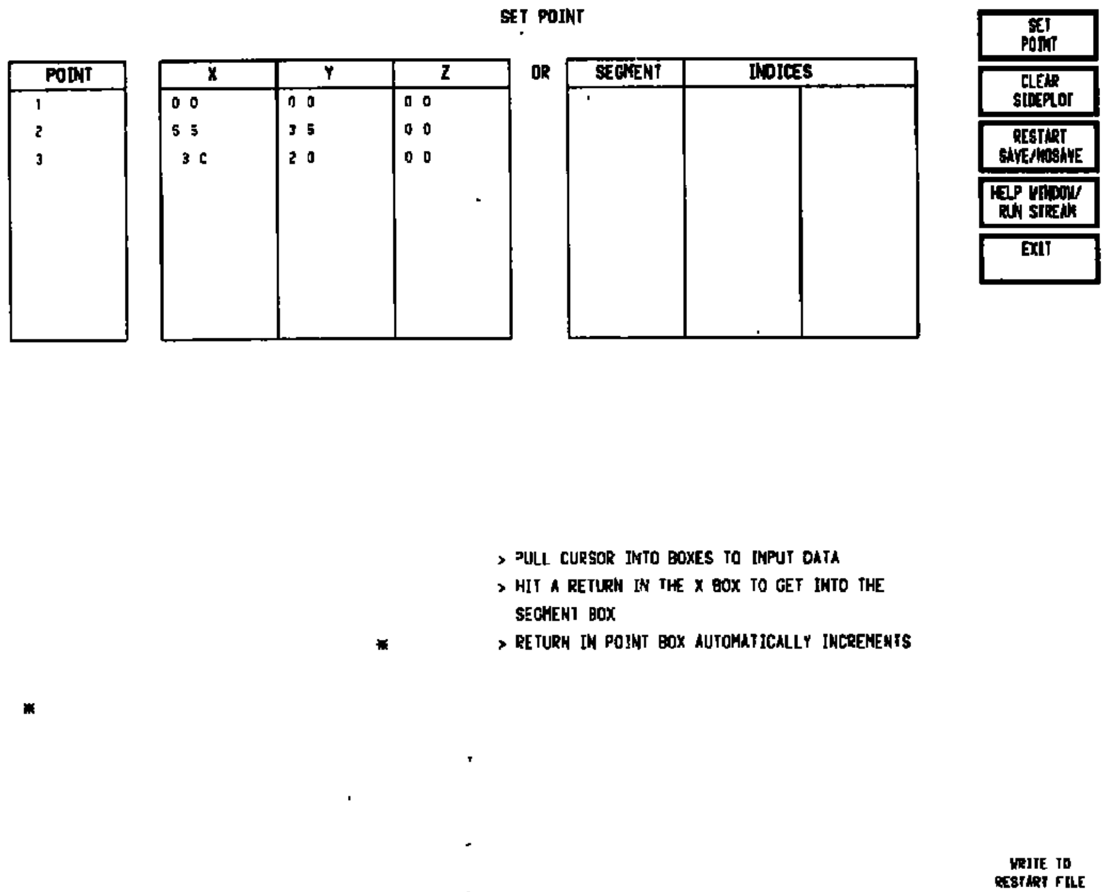


Figure 11. SET POINT form.

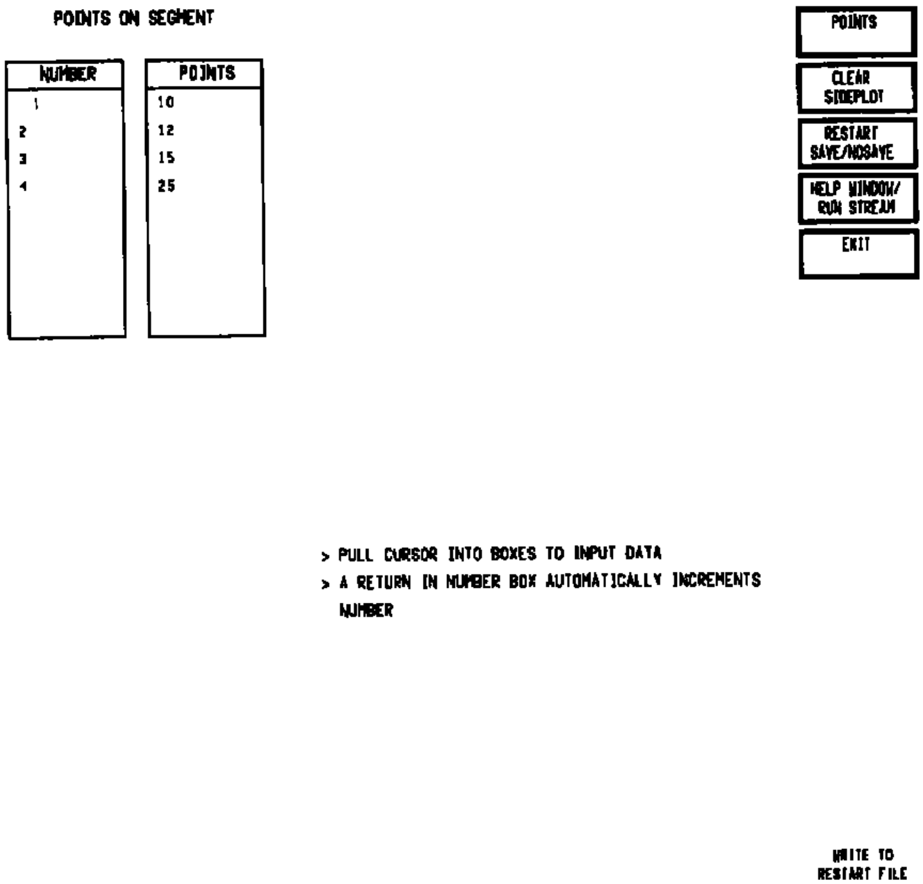


Figure 12. POINTS form.

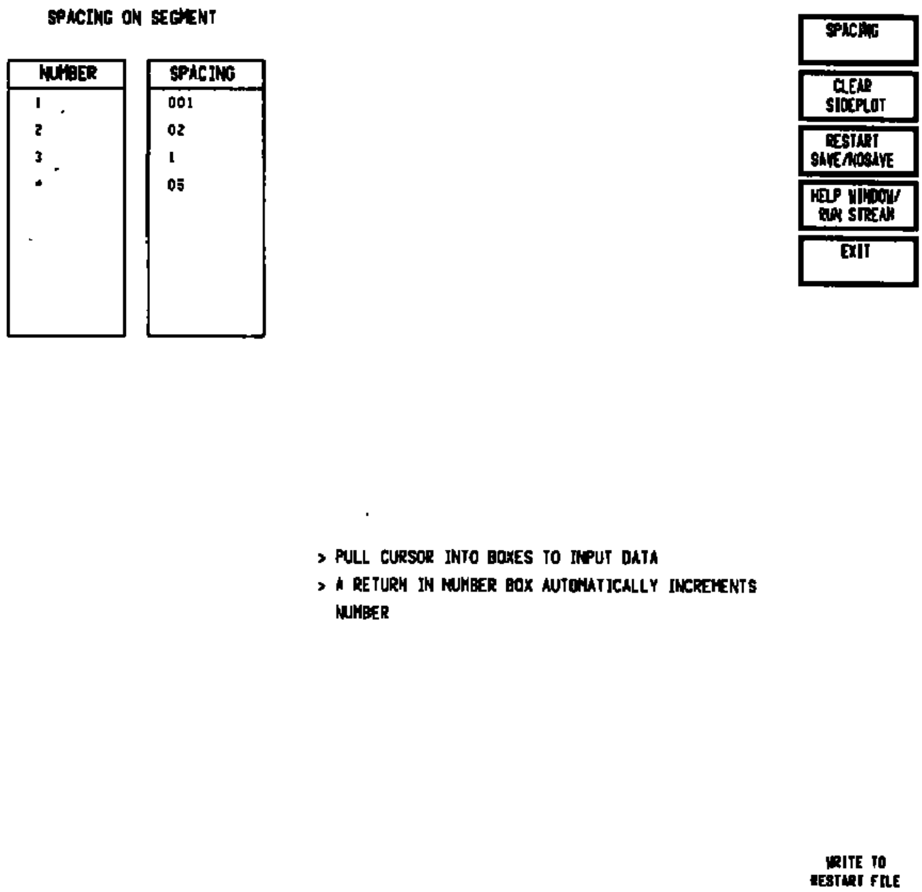


Figure 13. SPACING form.

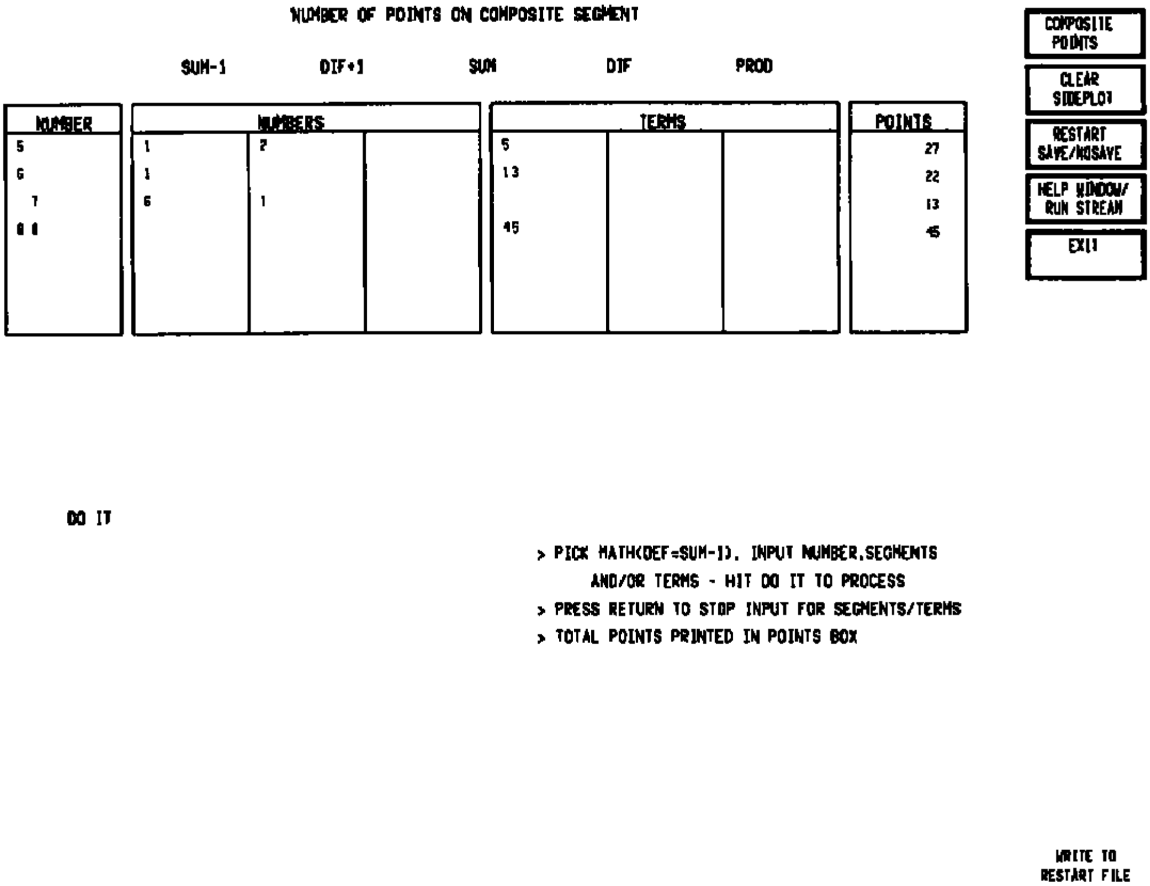


Figure 14. COMPOSITE POINTS form.

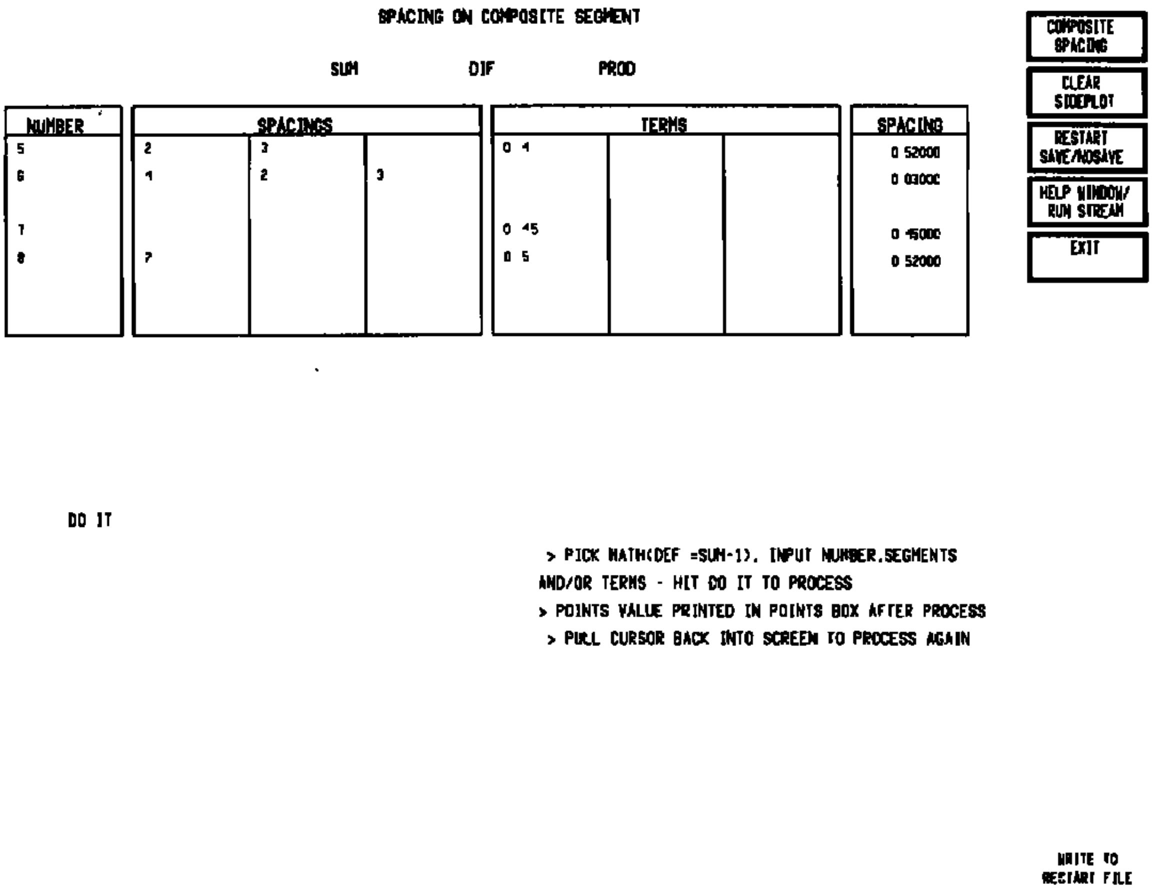


Figure 15. COMPOSITE SPACING form.

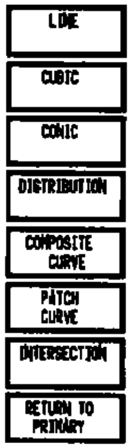


Figure 16. CURVES menu.

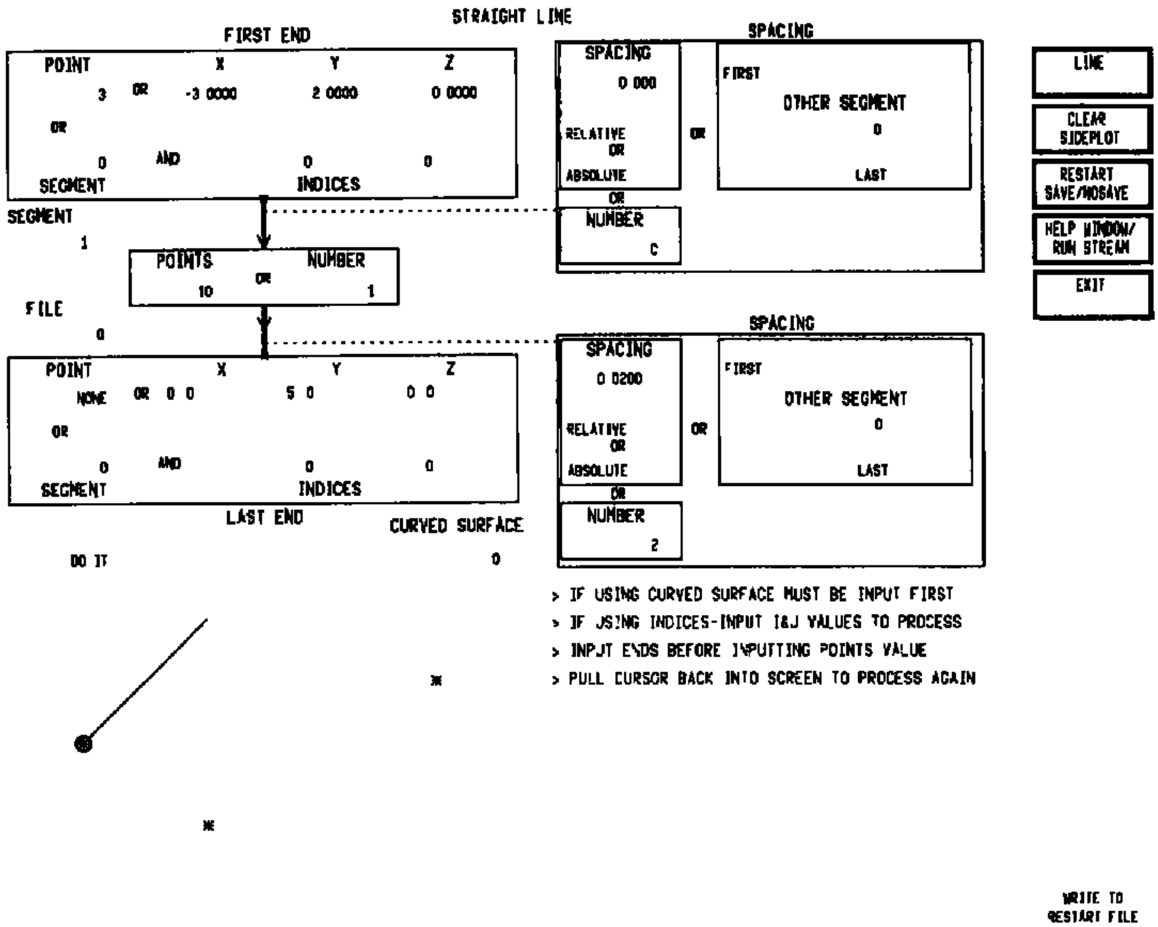


Figure 17. LINE form.

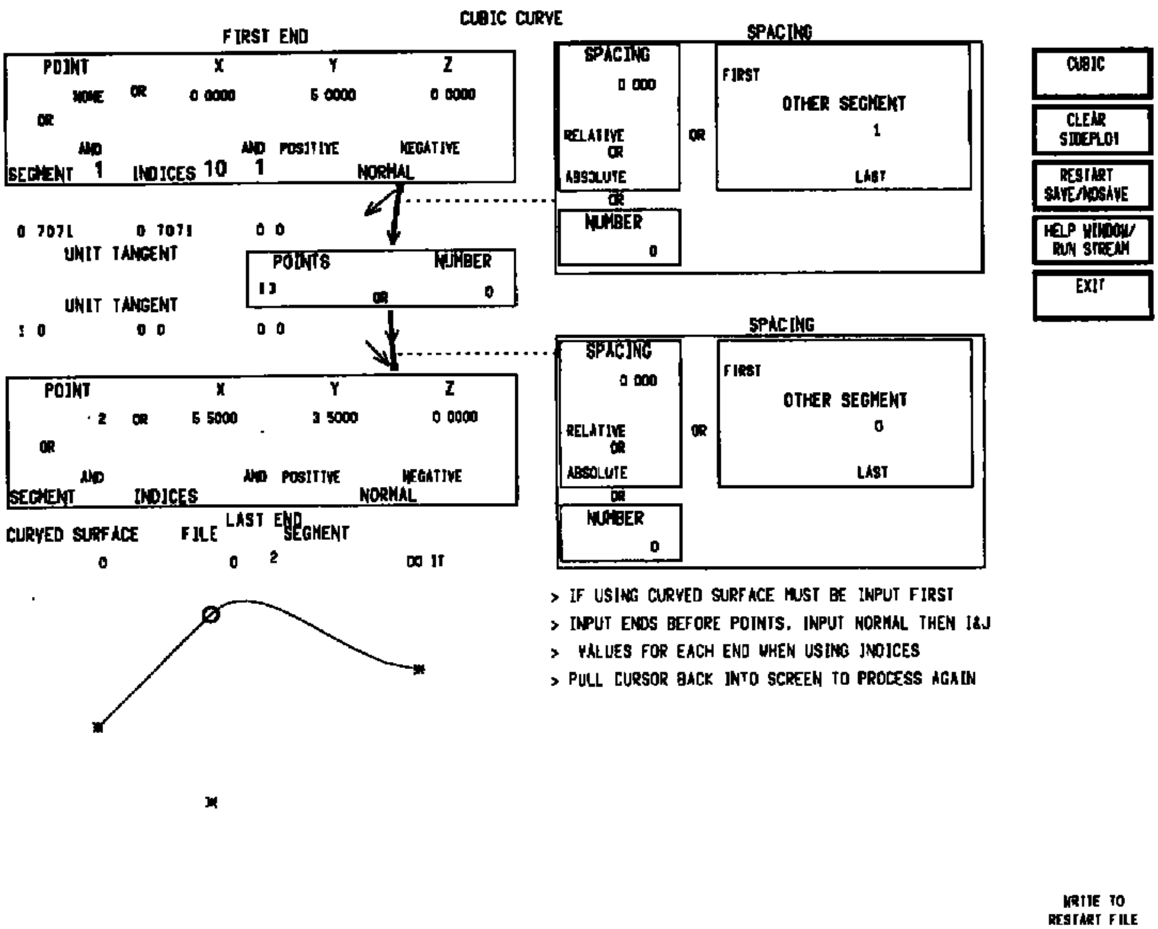


Figure 18. CUBIC form.

Figure 19. CONIC menu.

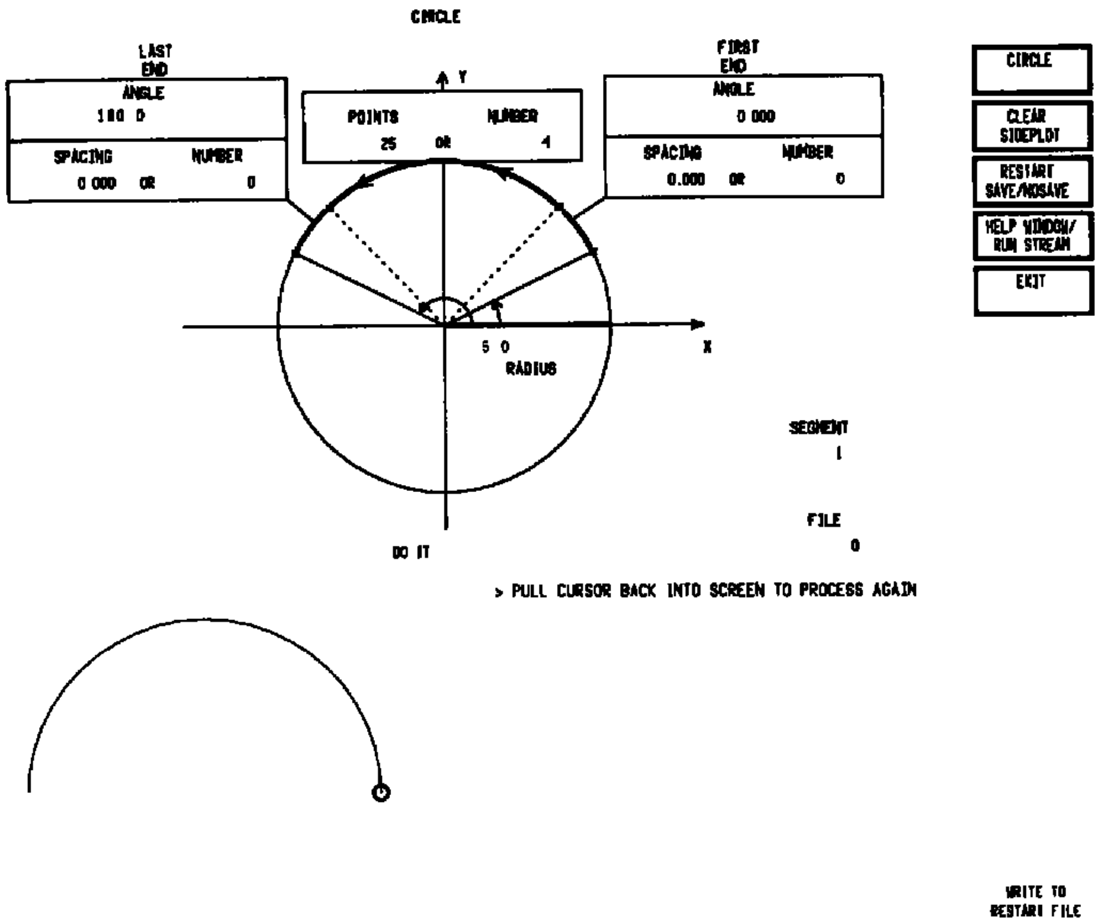


Figure 20. CIRCLE form.

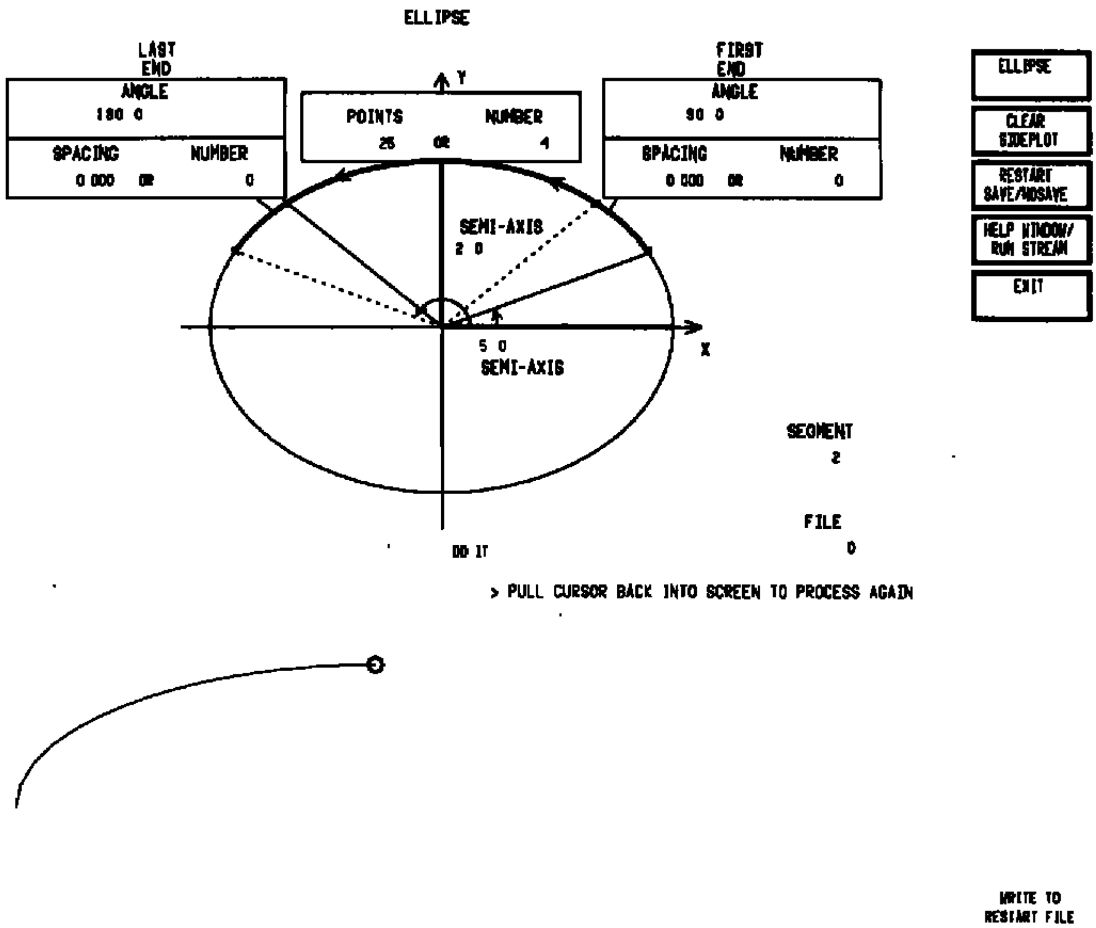


Figure 21. ELLIPSE form.

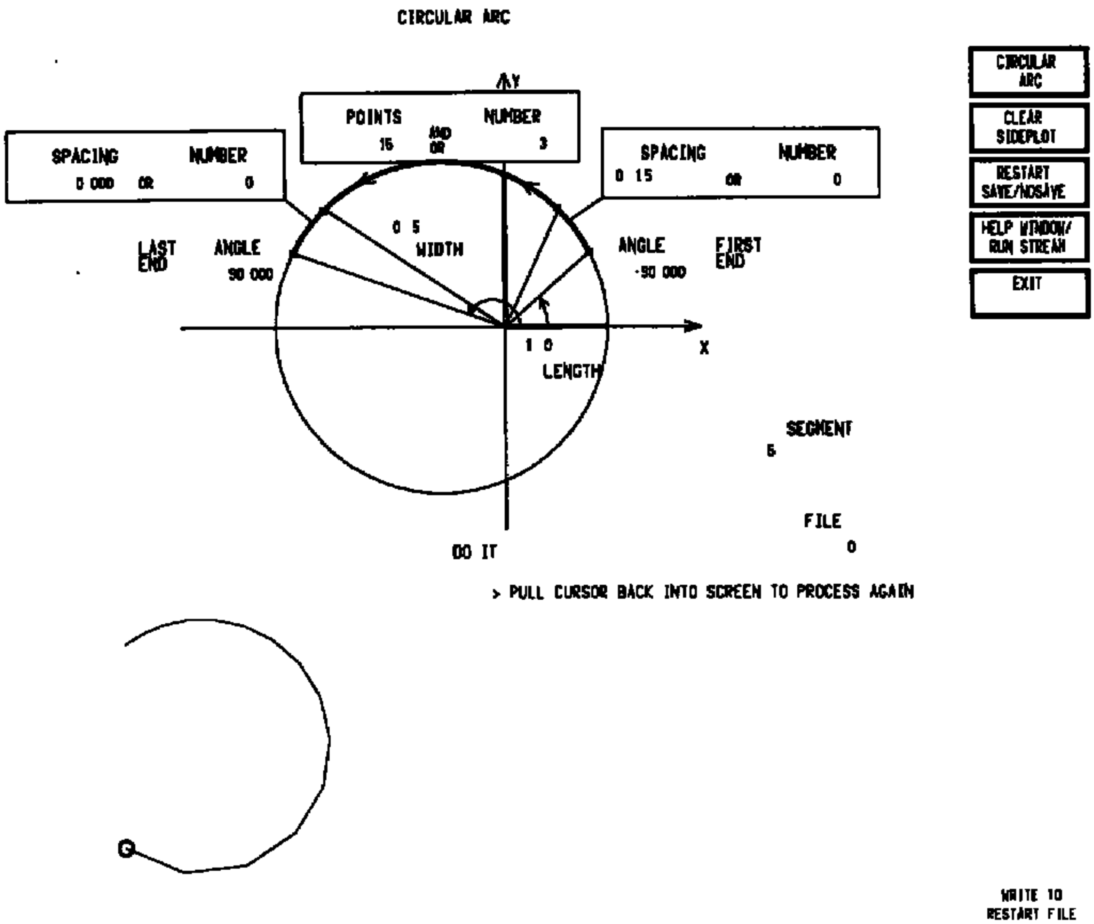


Figure 22. CIRCULAR ARC form.

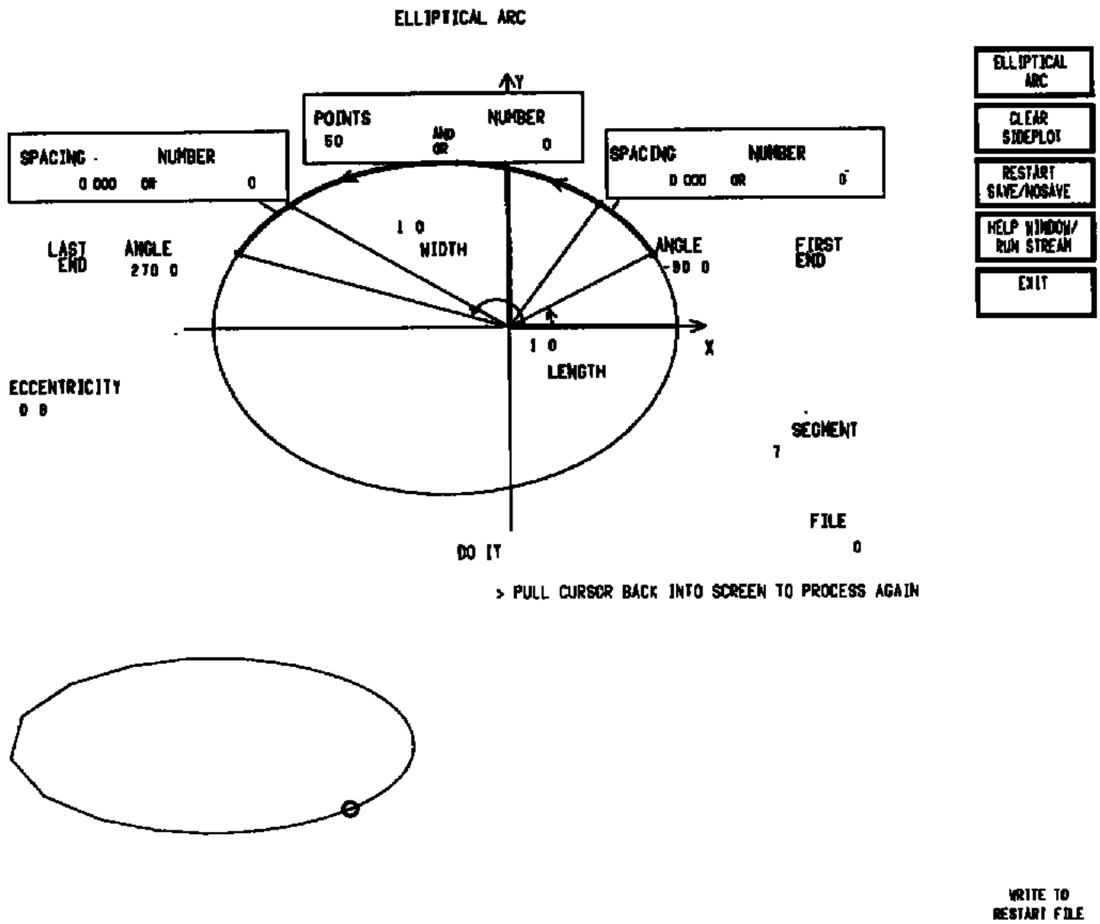


Figure 23. ELLIPTICAL ARC form.

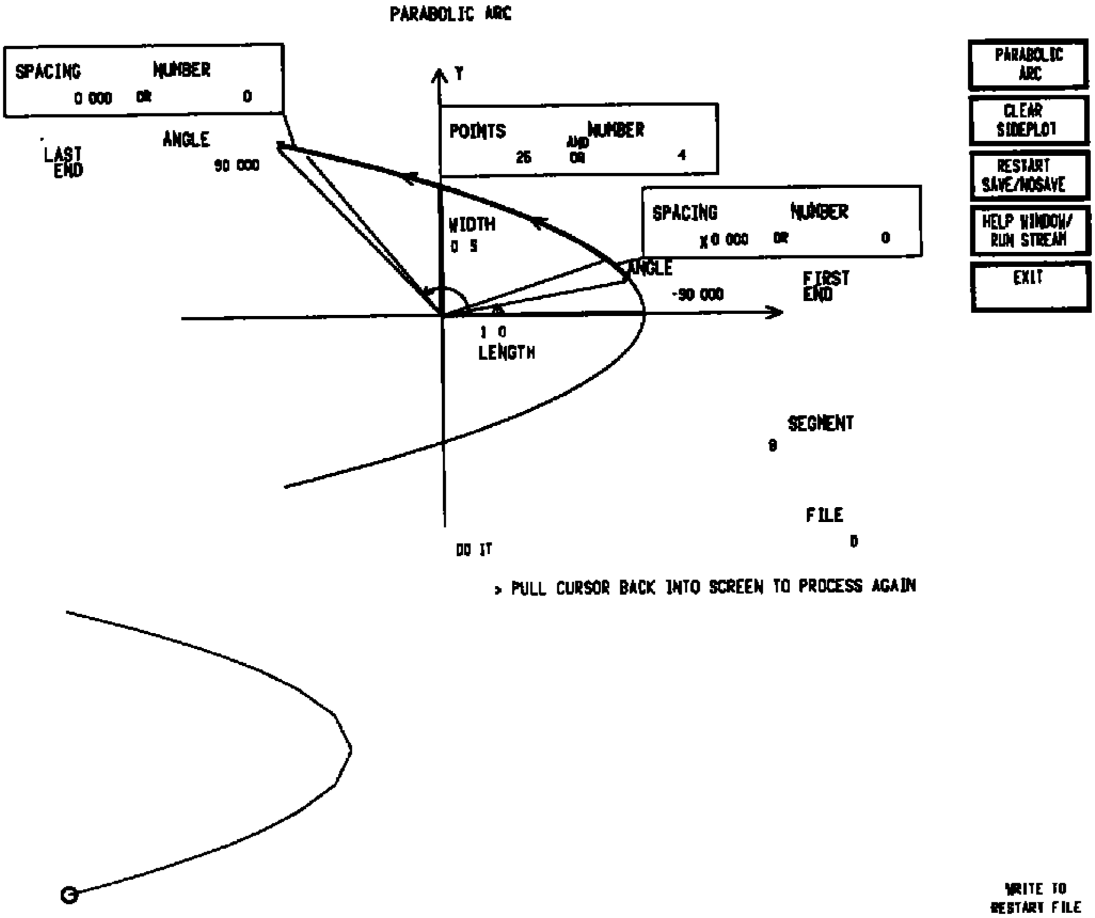


Figure 24. PARABOLIC ARC form.

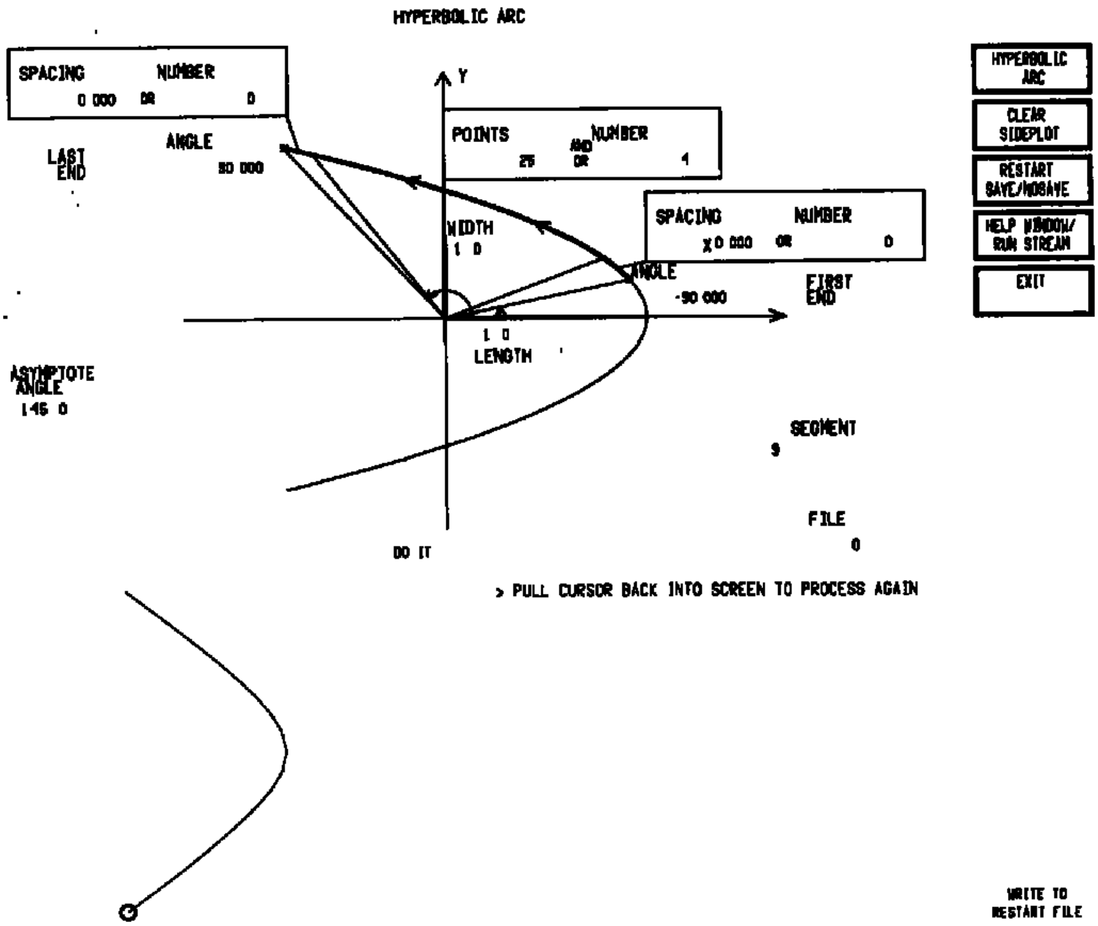


Figure 25. HYPERBOLIC ARC form.

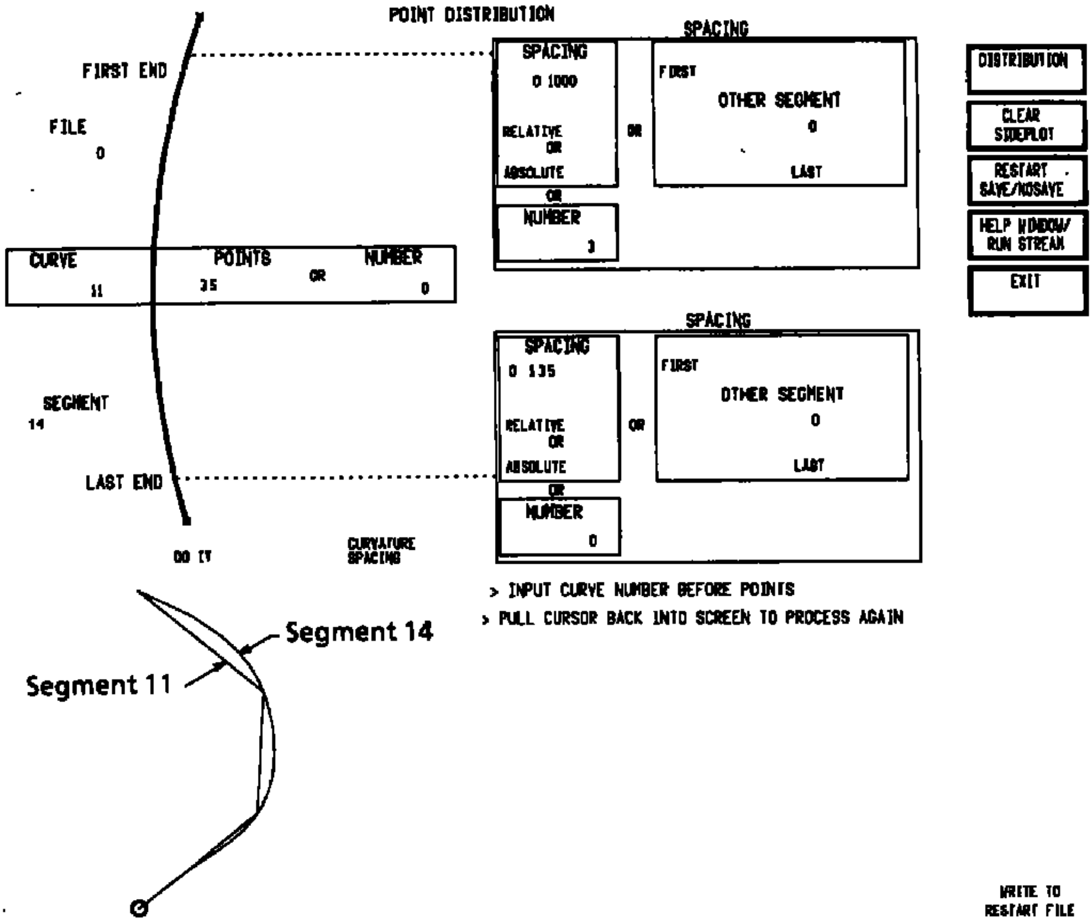


Figure 26. DISTRIBUTION form.

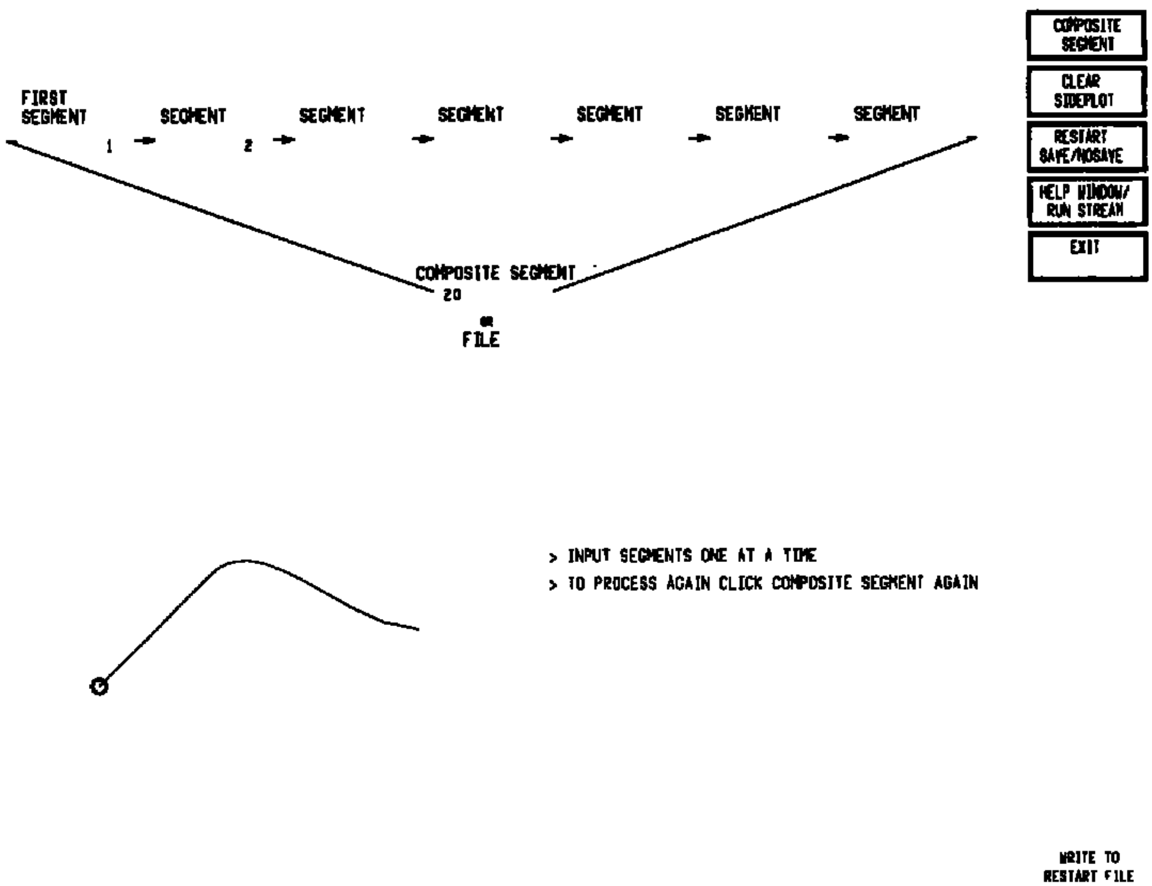


Figure 27. COMPOSITE CURVE form.

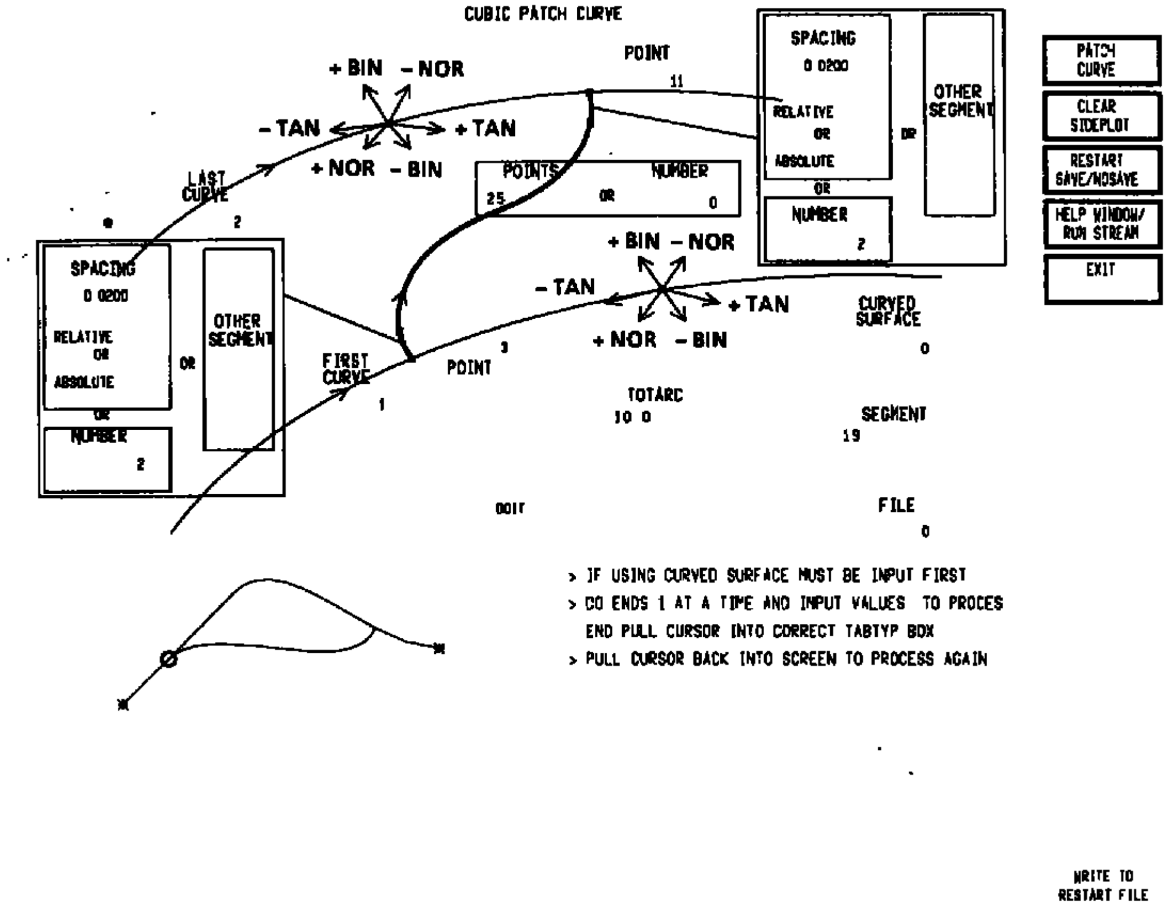


Figure 28. PATCH CURVE form.

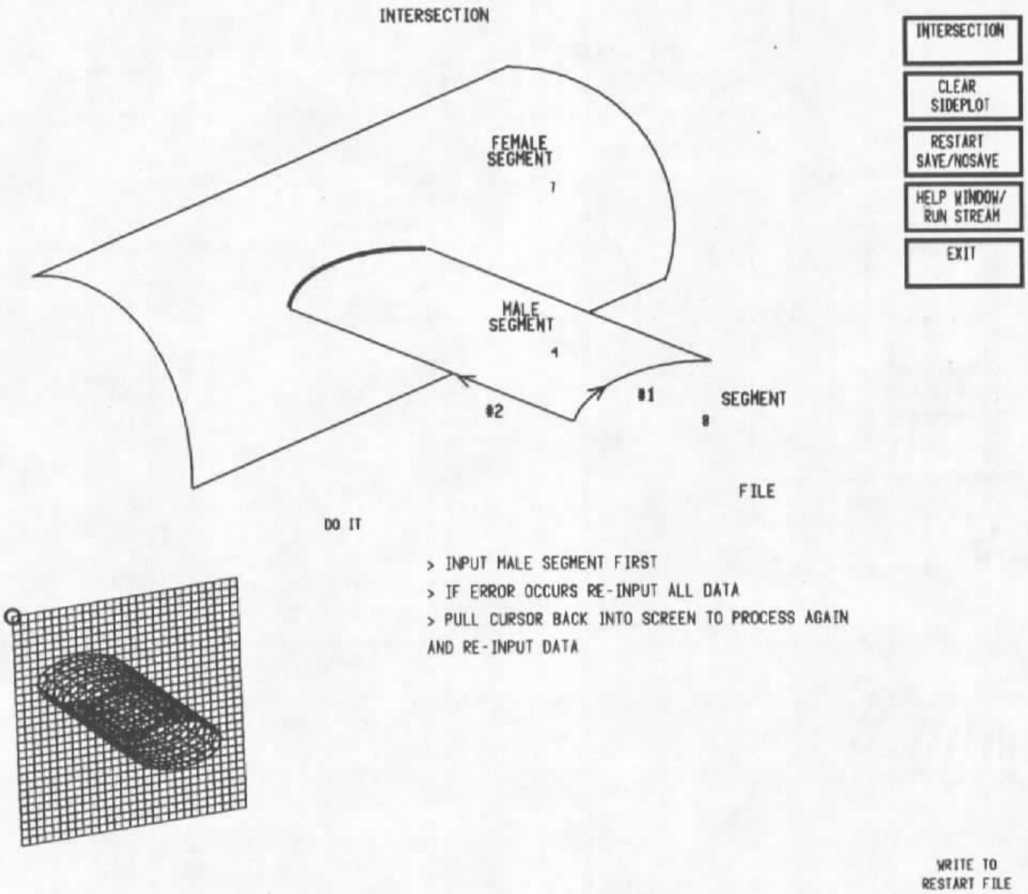


Figure 29. INTERSECTION form.

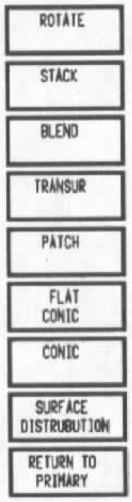


Figure 30. SURFACES menu.

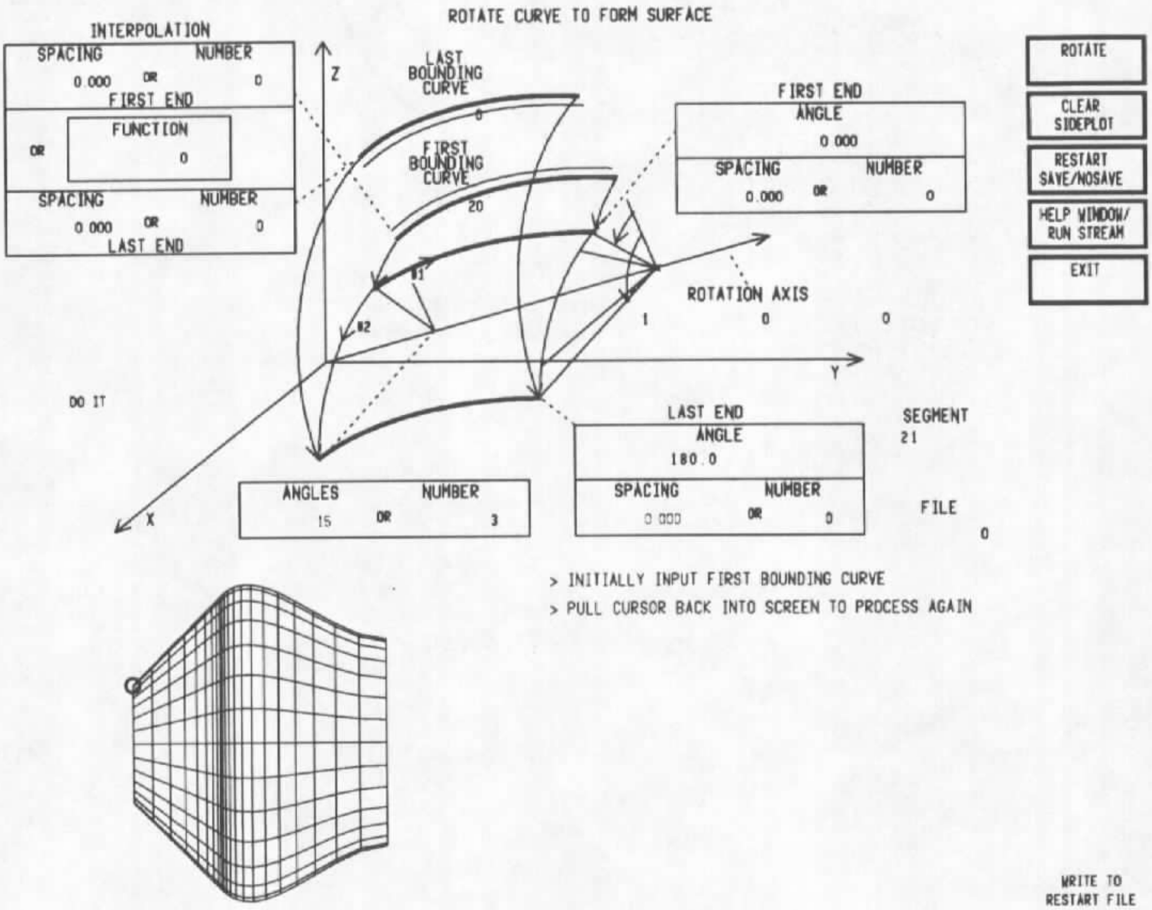


Figure 31. ROTATE form.

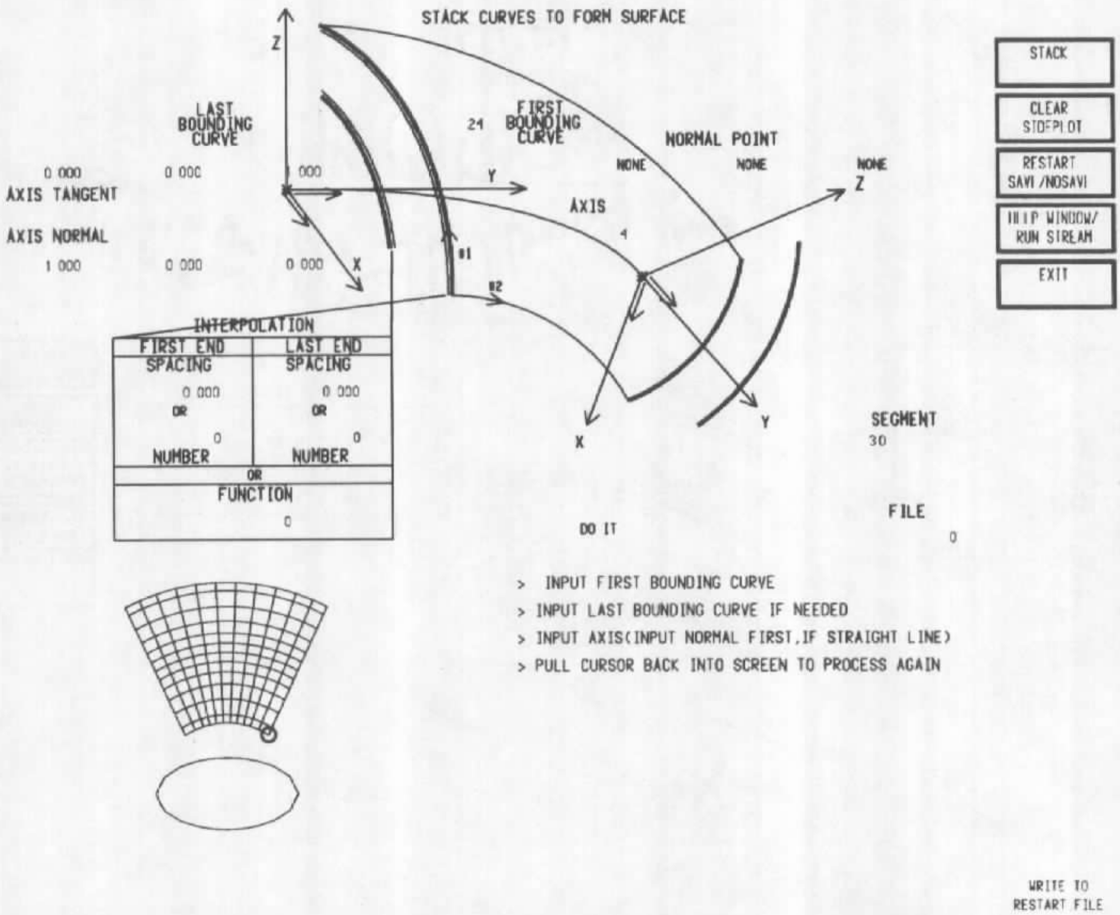


Figure 32. STACK form.

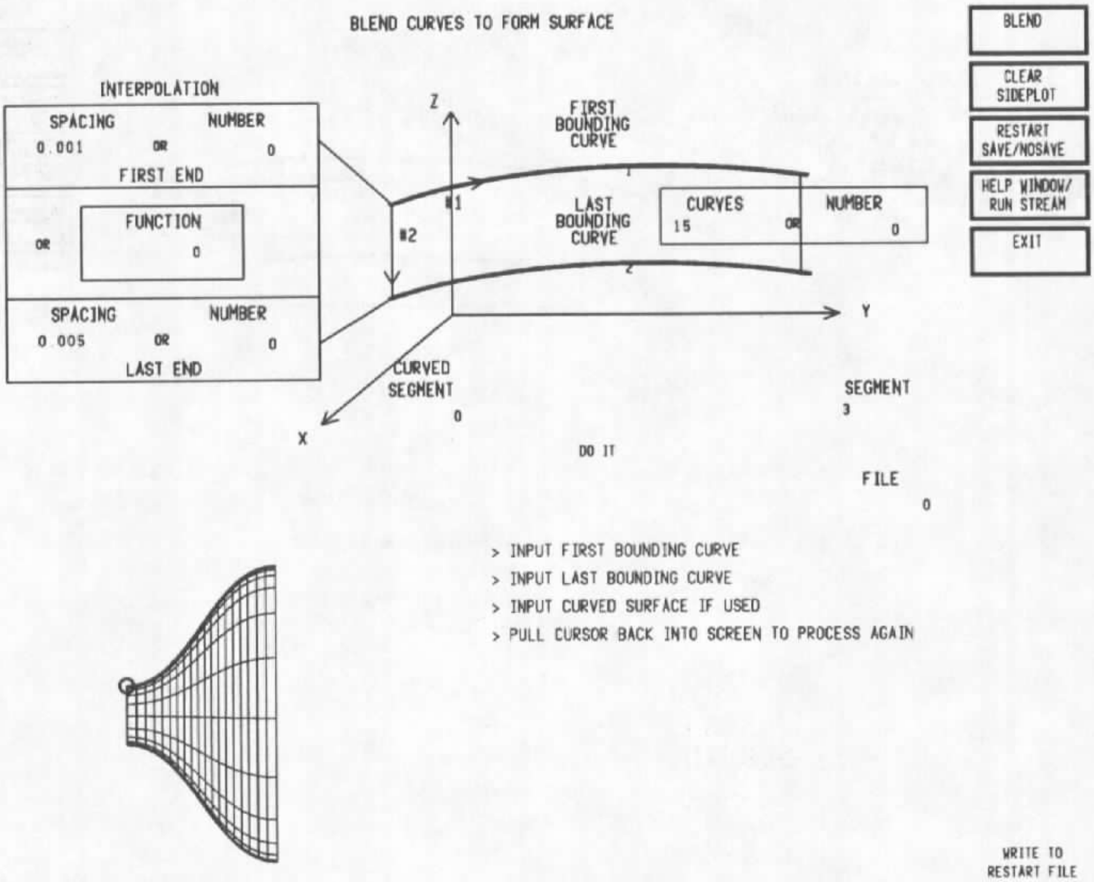


Figure 33. BLEND form.

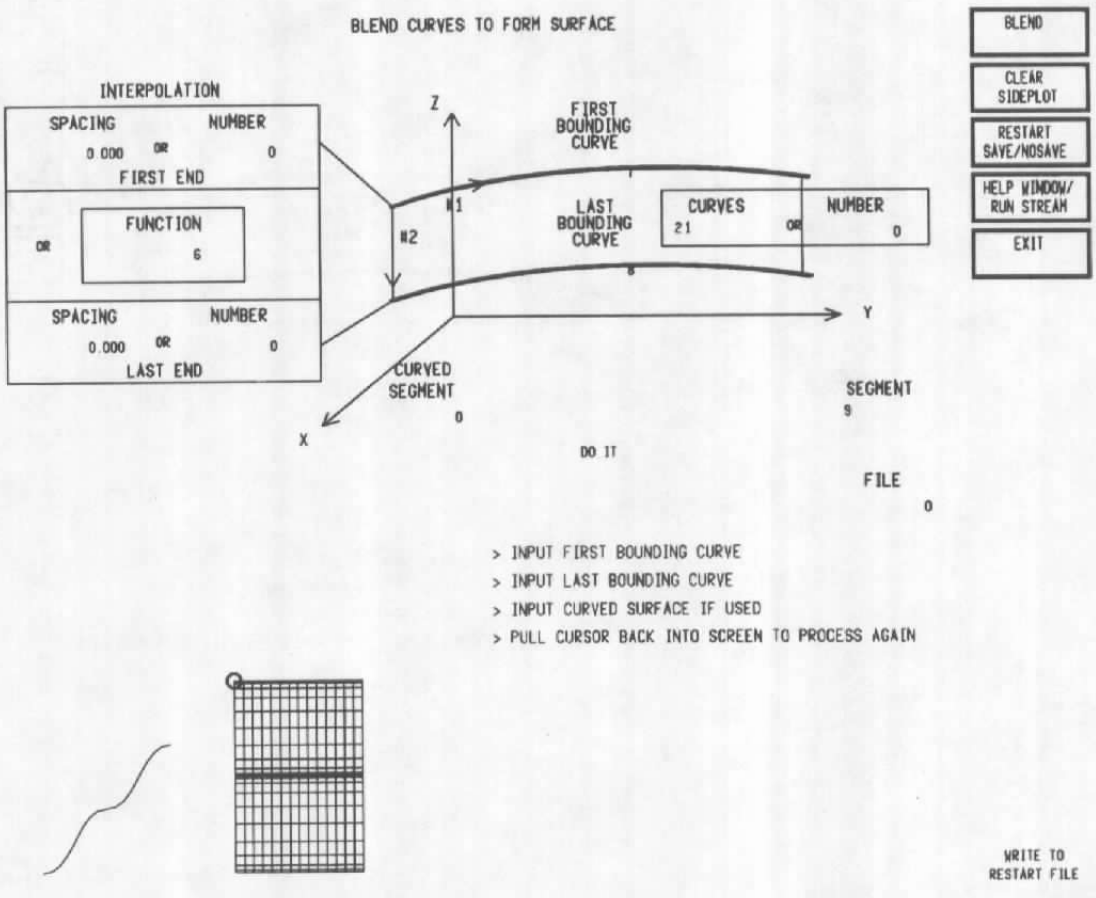


Figure 34. Spacing using a function.

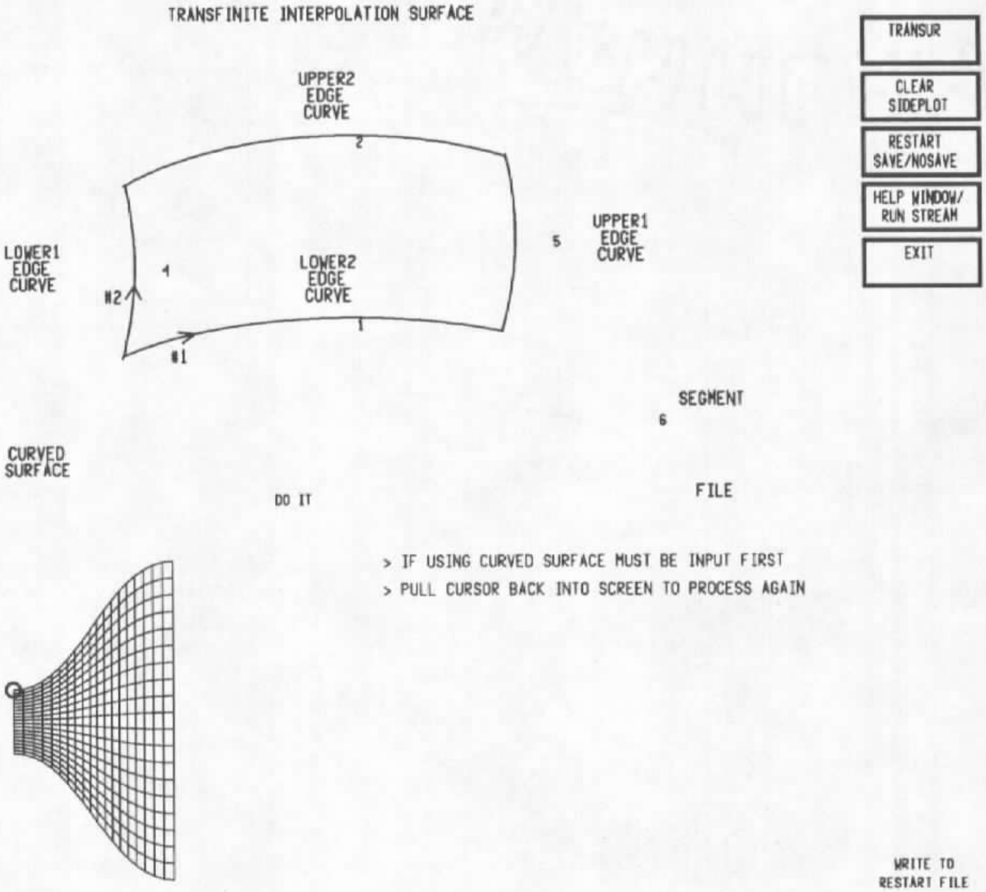


Figure 35. TRANSUR form.

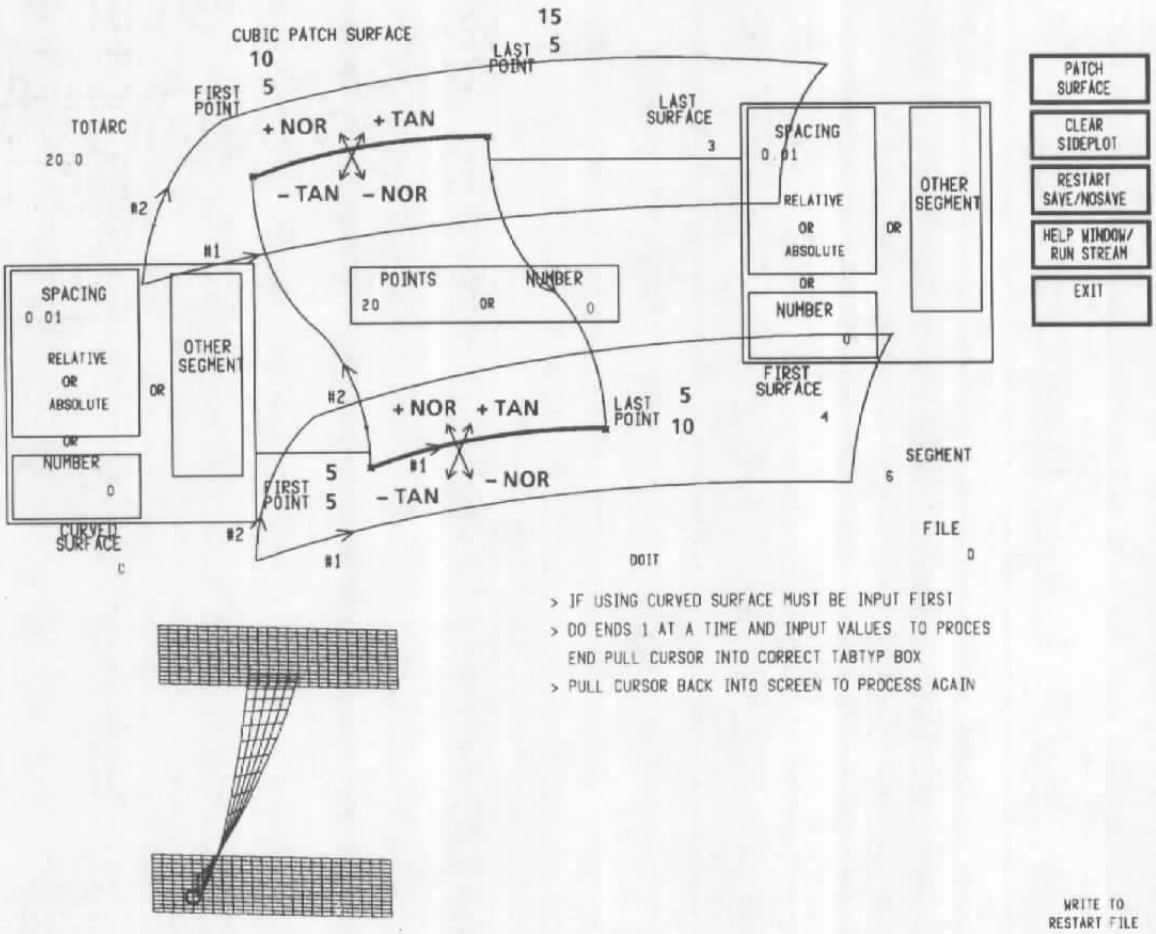


Figure 36. PATCH form.

Figure 37. FLAT CONIC menu.

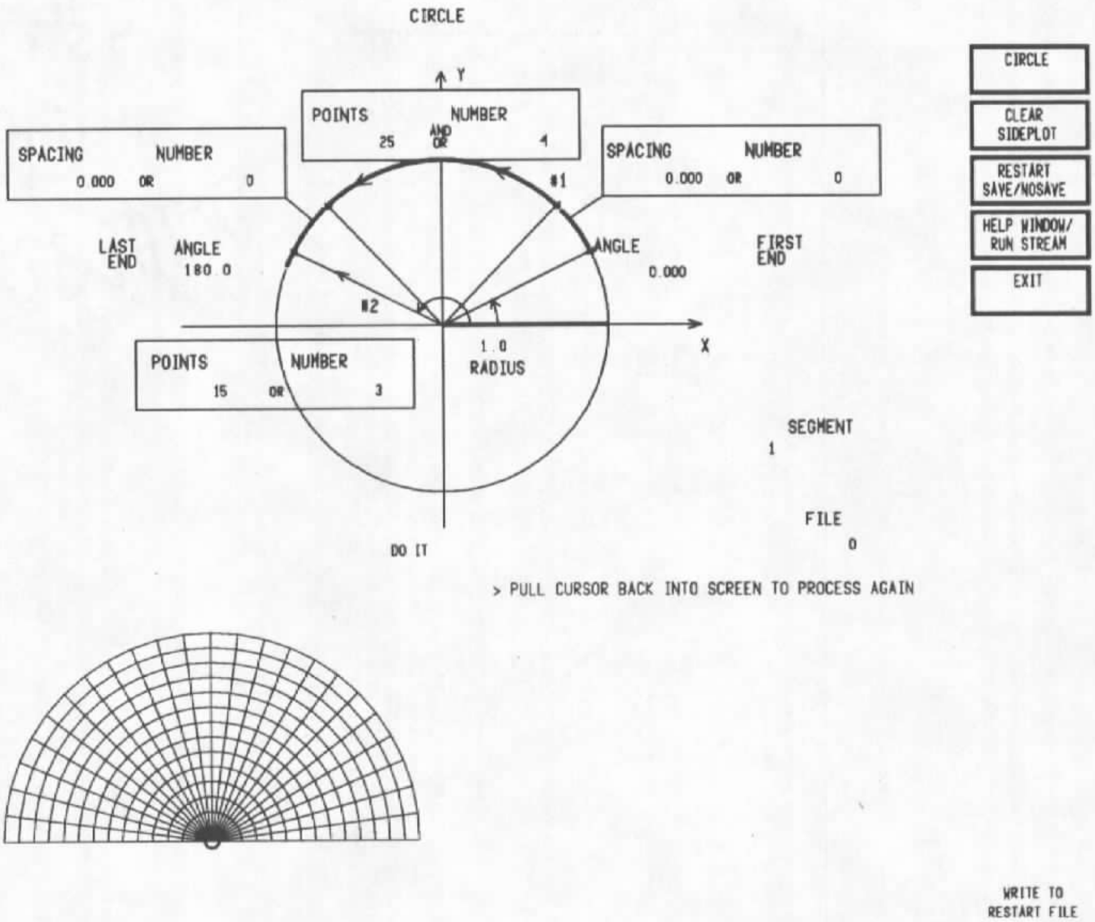


Figure 38. CIRCLE form.

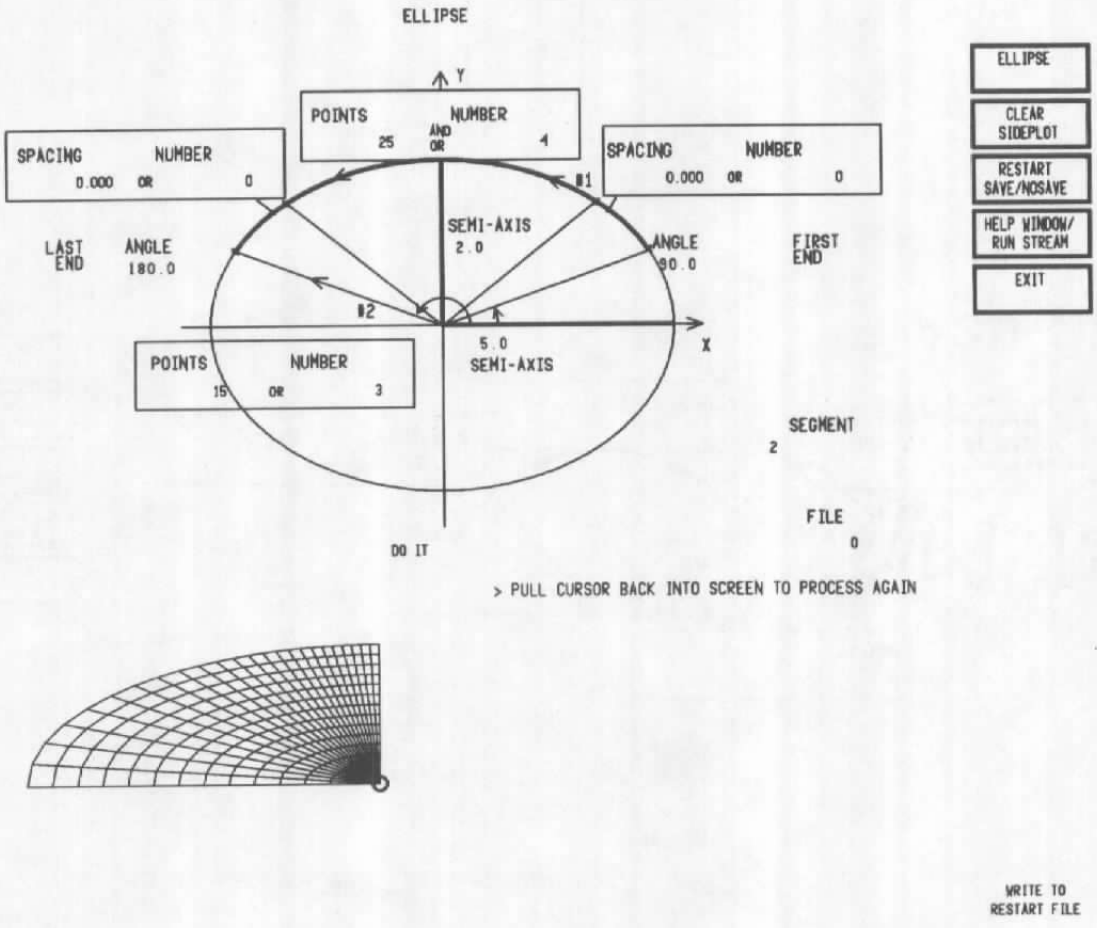


Figure 39. ELLIPSE form.

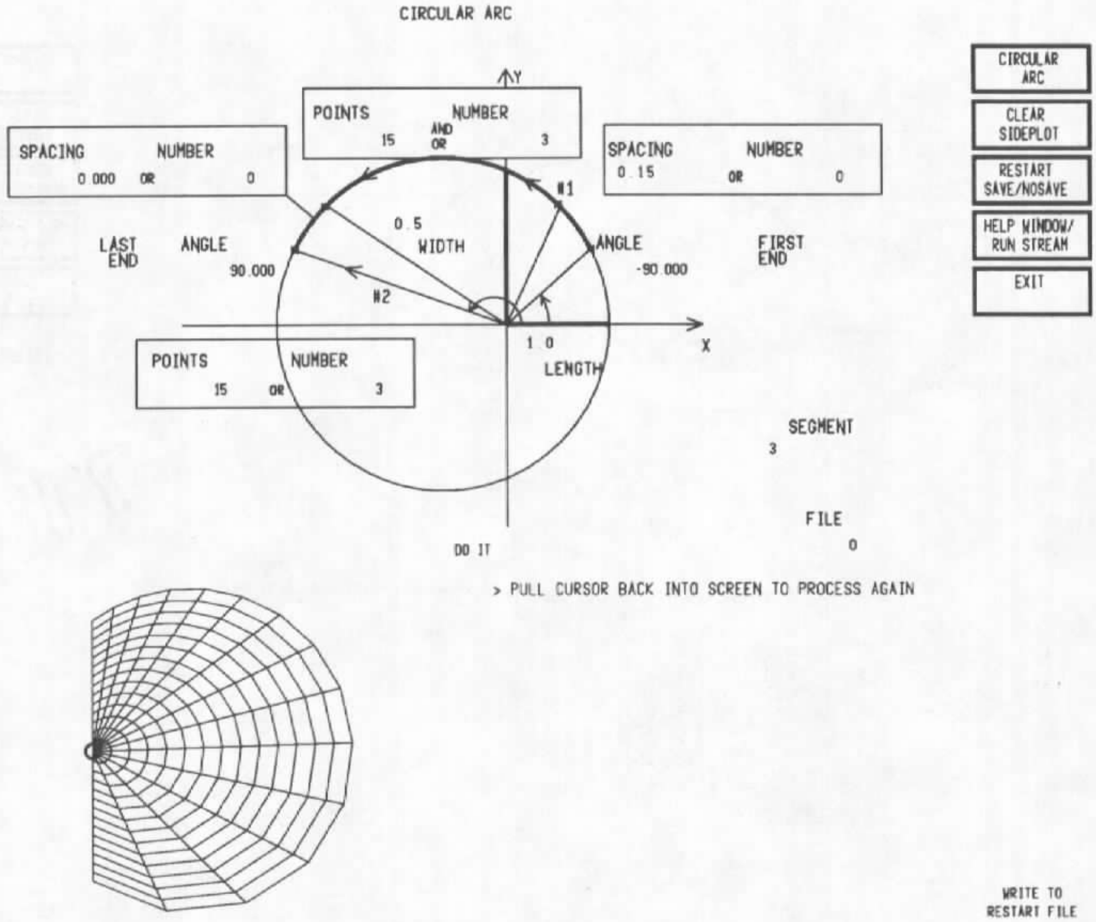


Figure 40. CIRCULAR ARC form.

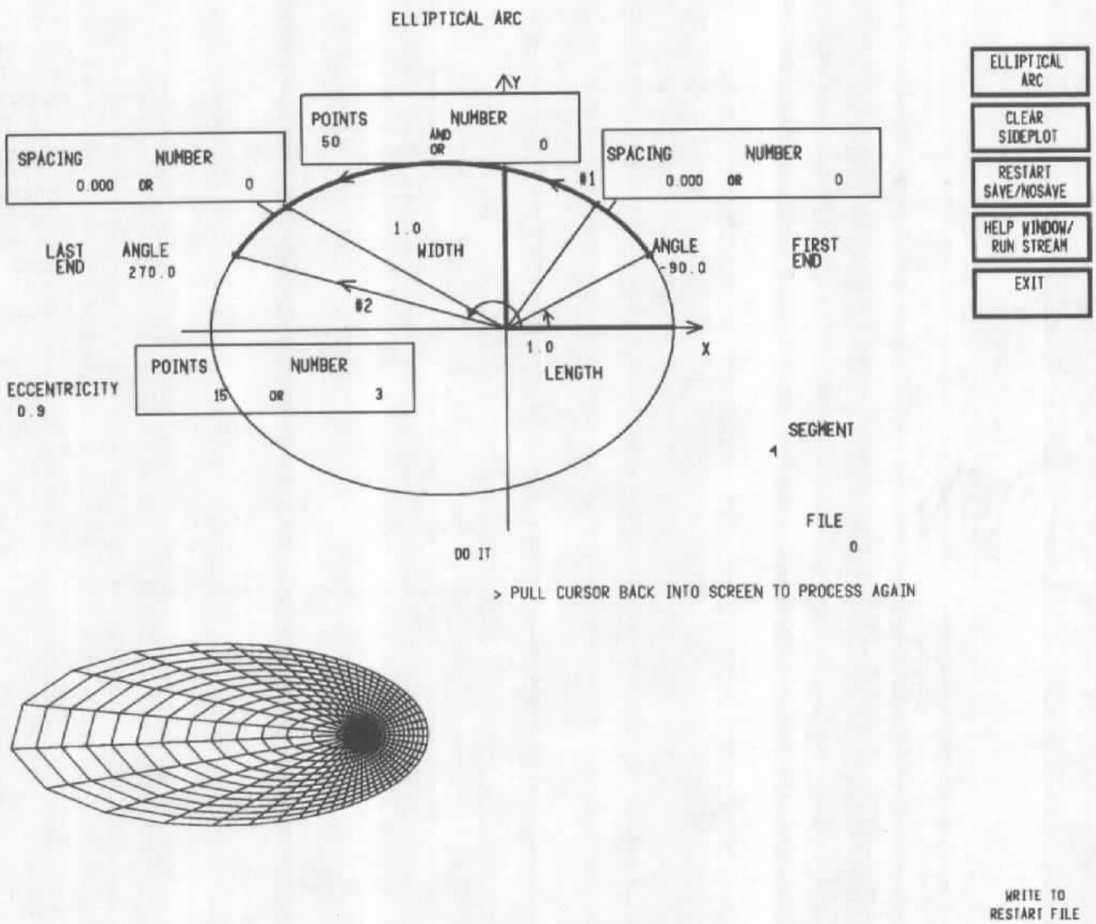


Figure 41. ELLIPTICAL ARC form.

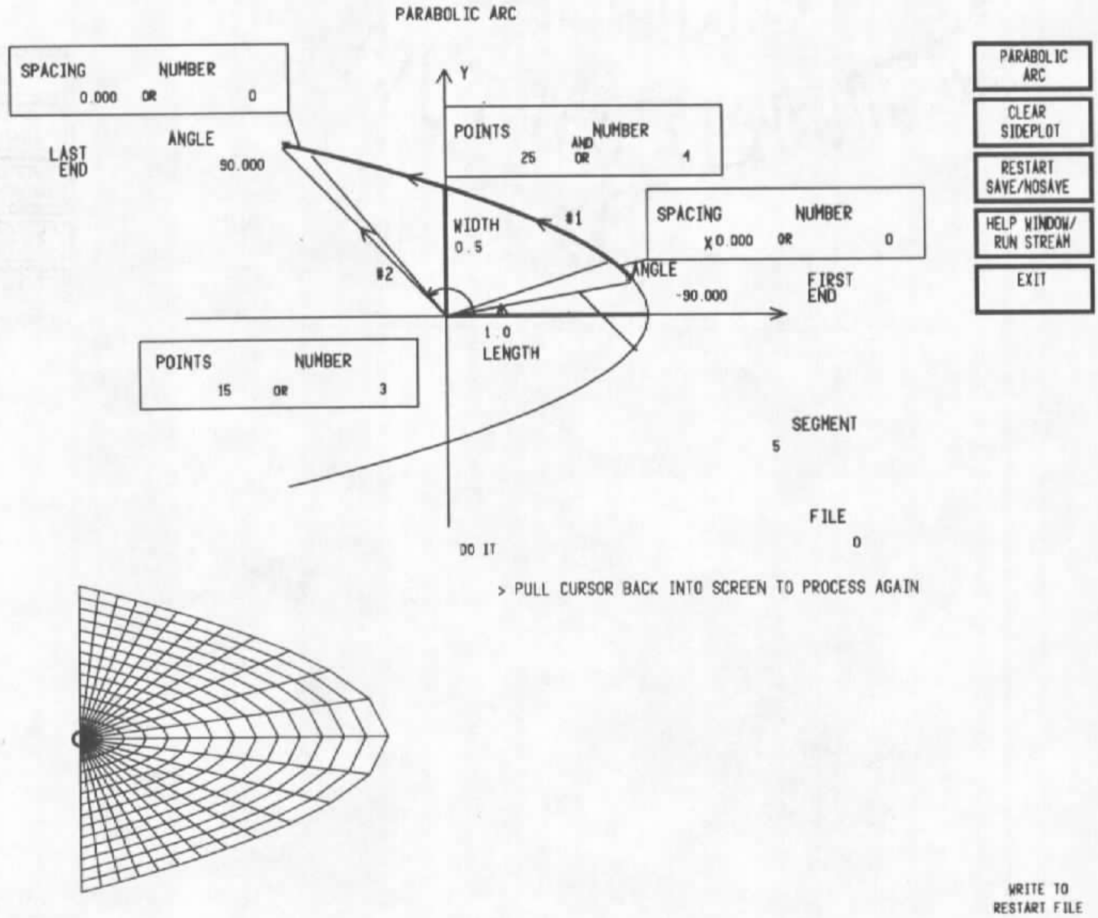


Figure 42. PARABOLIC ARC form.

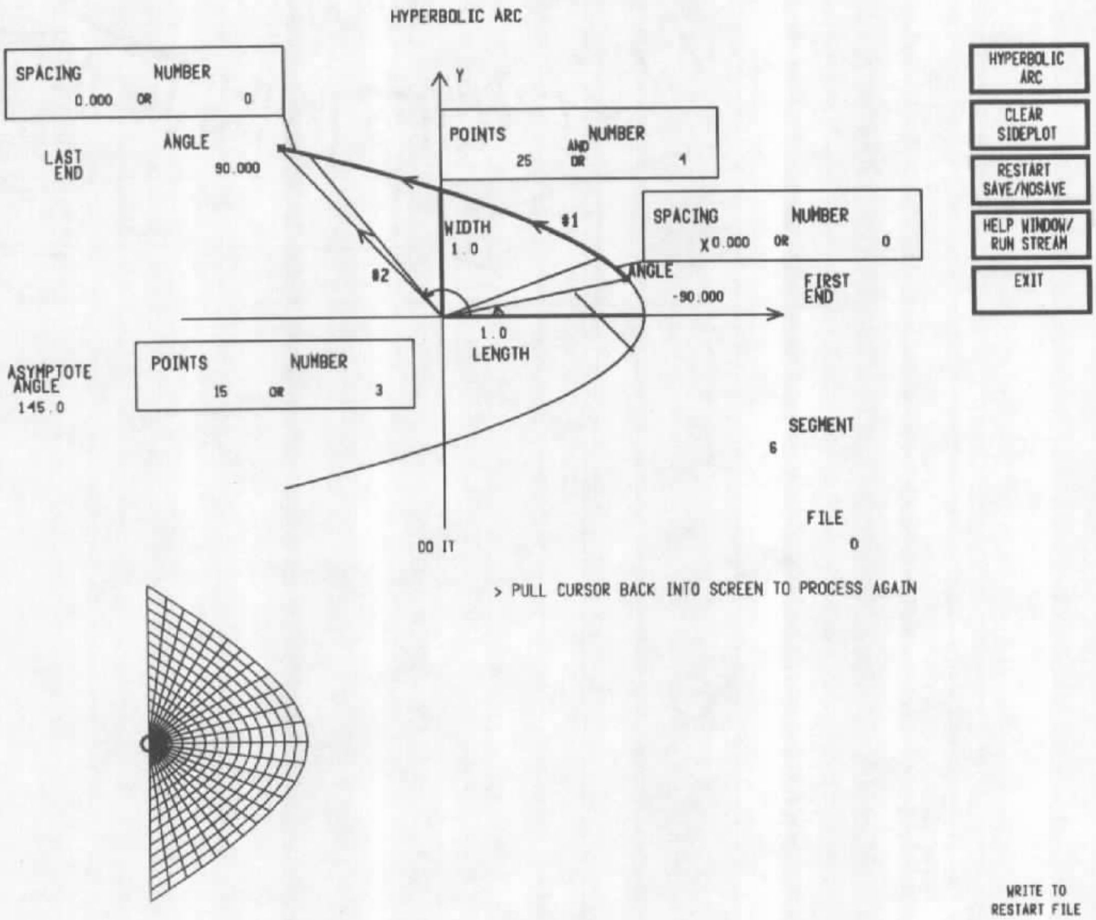


Figure 43. HYPERBOLIC ARC form.

Figure 44. CONIC menu.

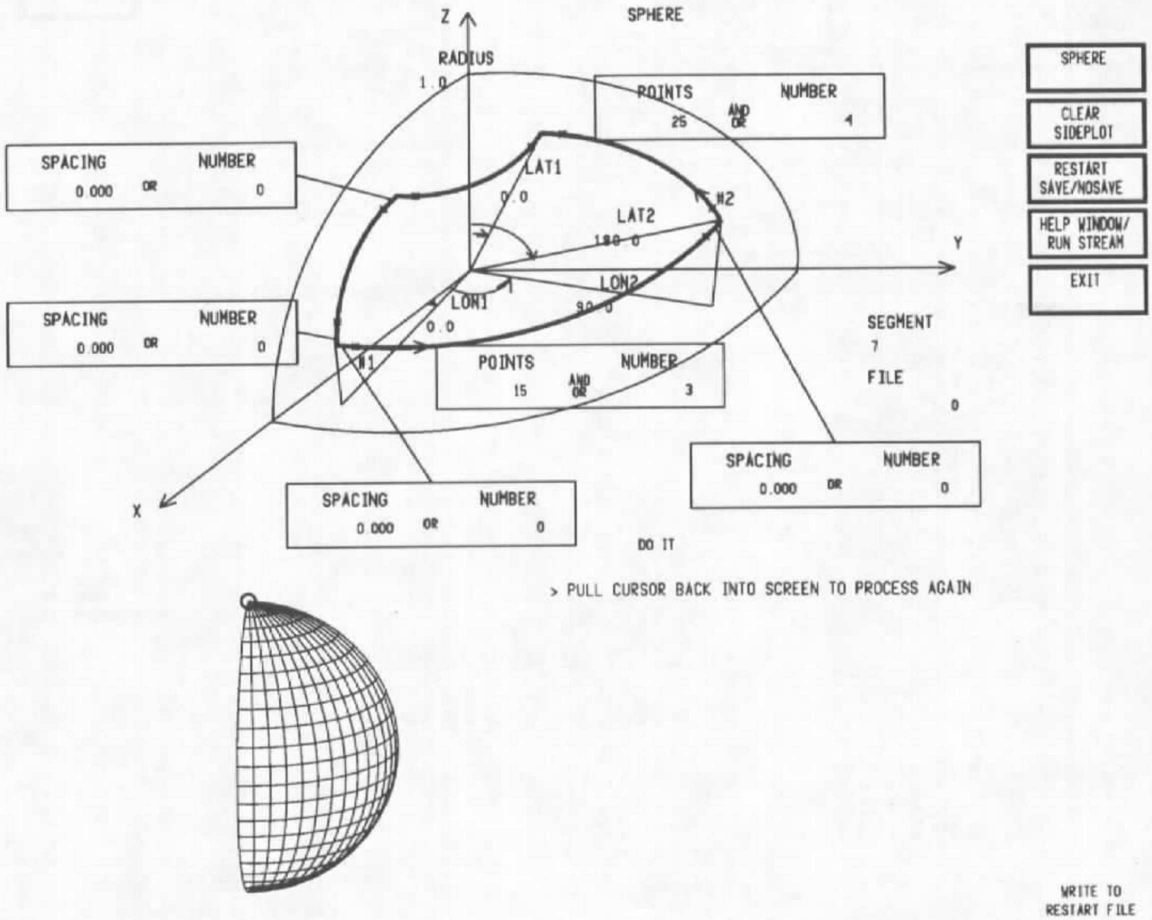


Figure 45. SPHERE form.

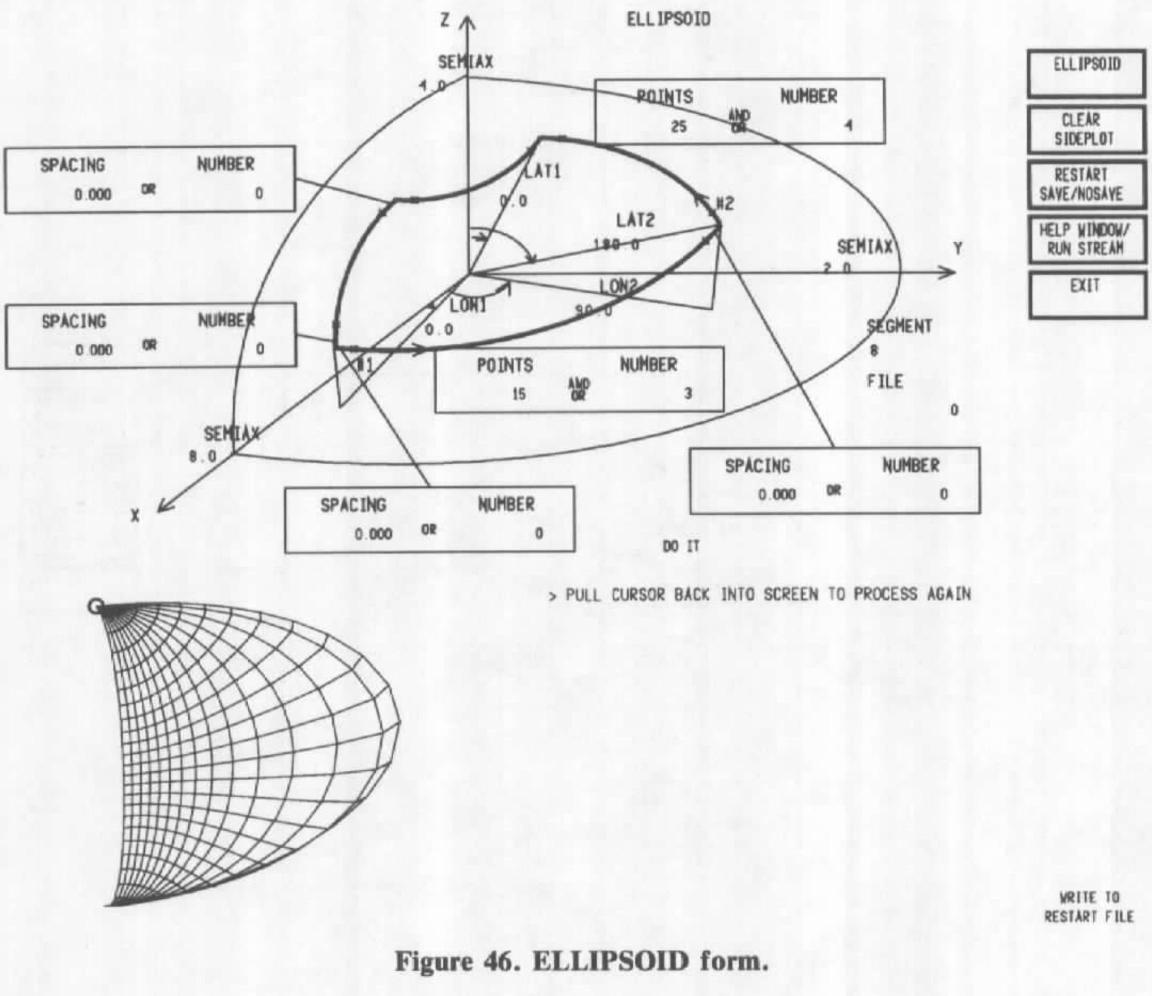


Figure 46. ELLIPSOID form.

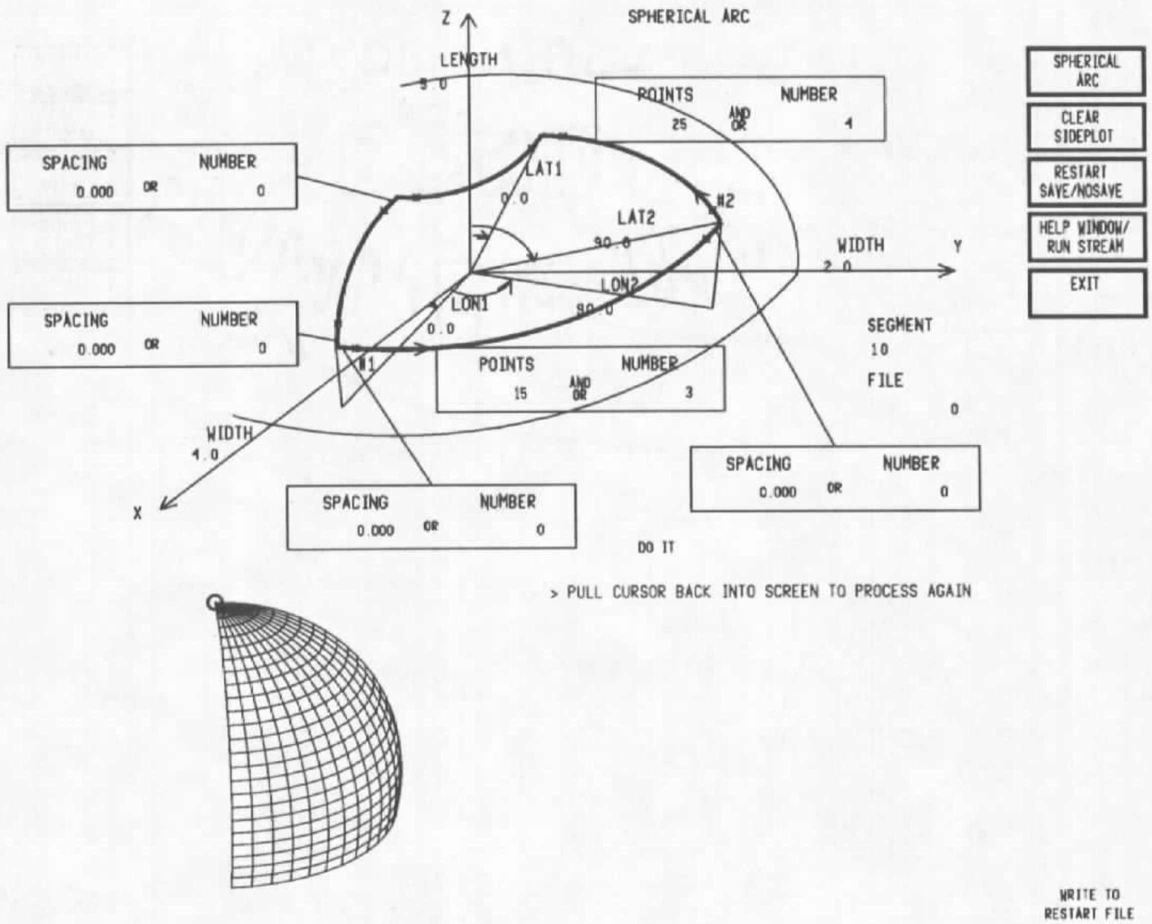


Figure 47. SPHERICAL ARC form.

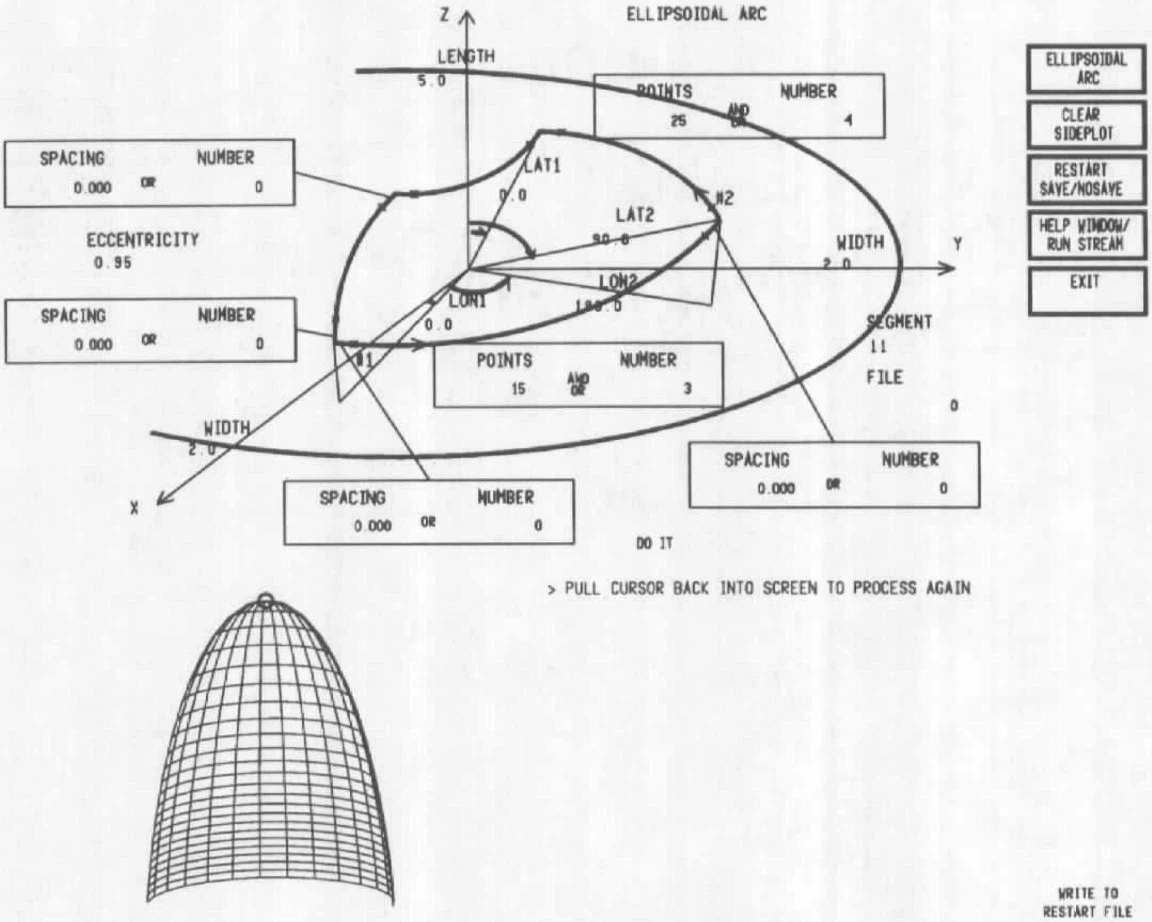


Figure 48. ELLIPSOIDAL ARC form.

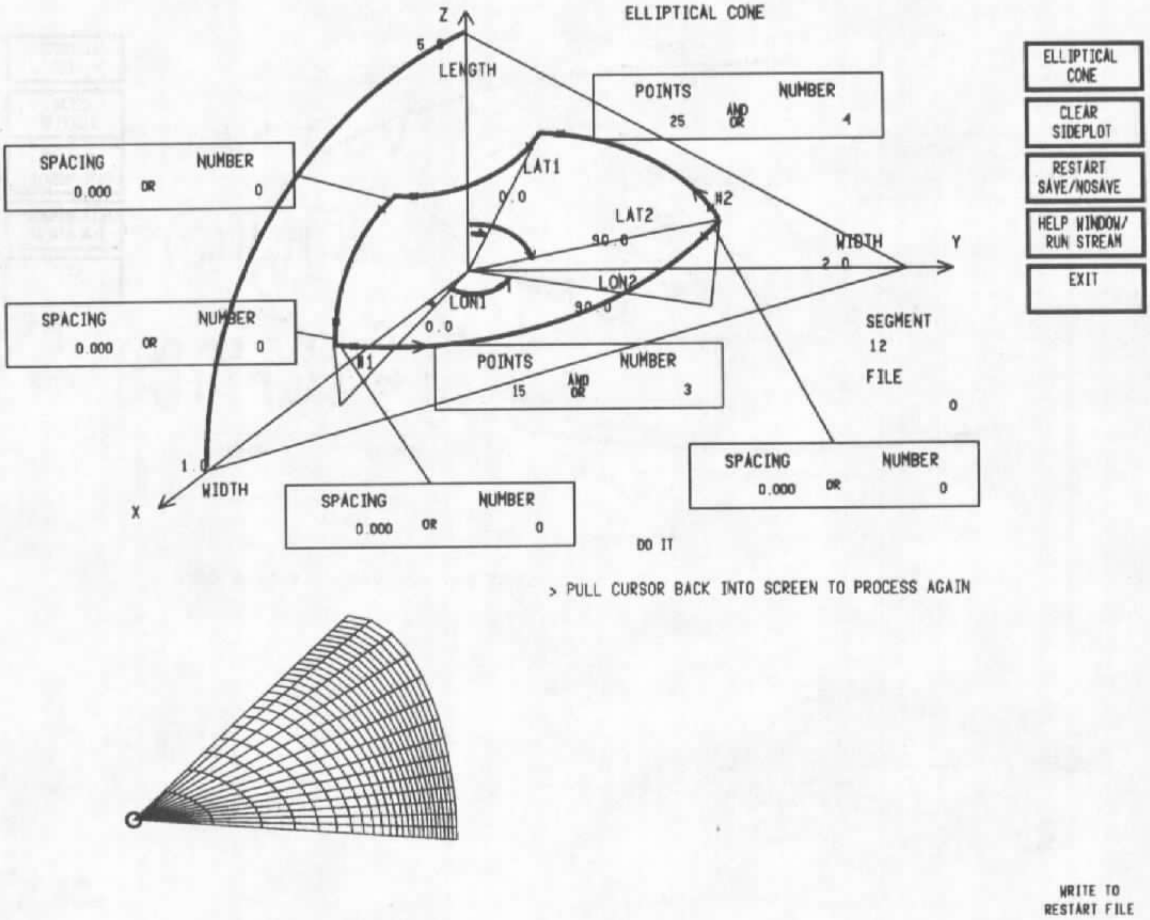


Figure 49. ELLIPTICAL CONE form.

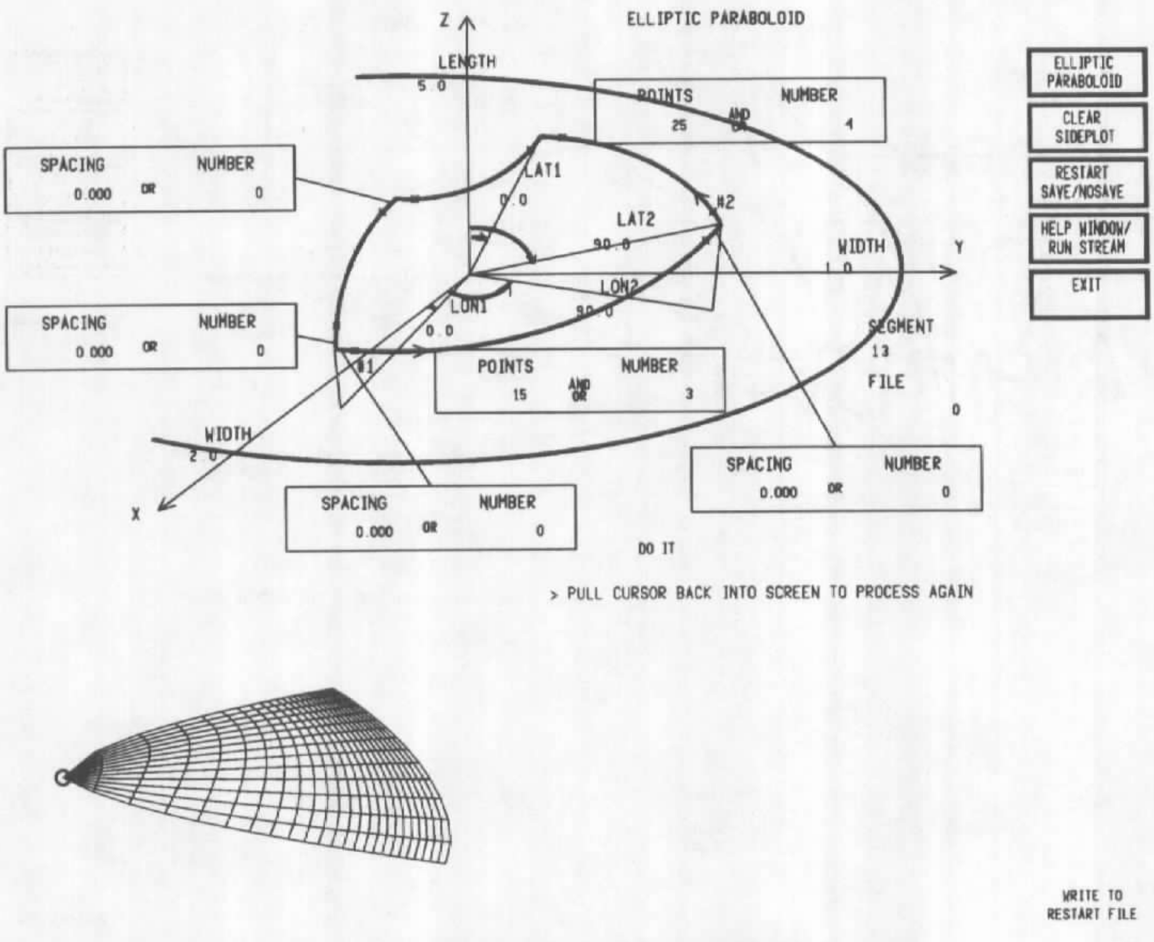


Figure 50. ELLIPTIC PARABOLOID form.

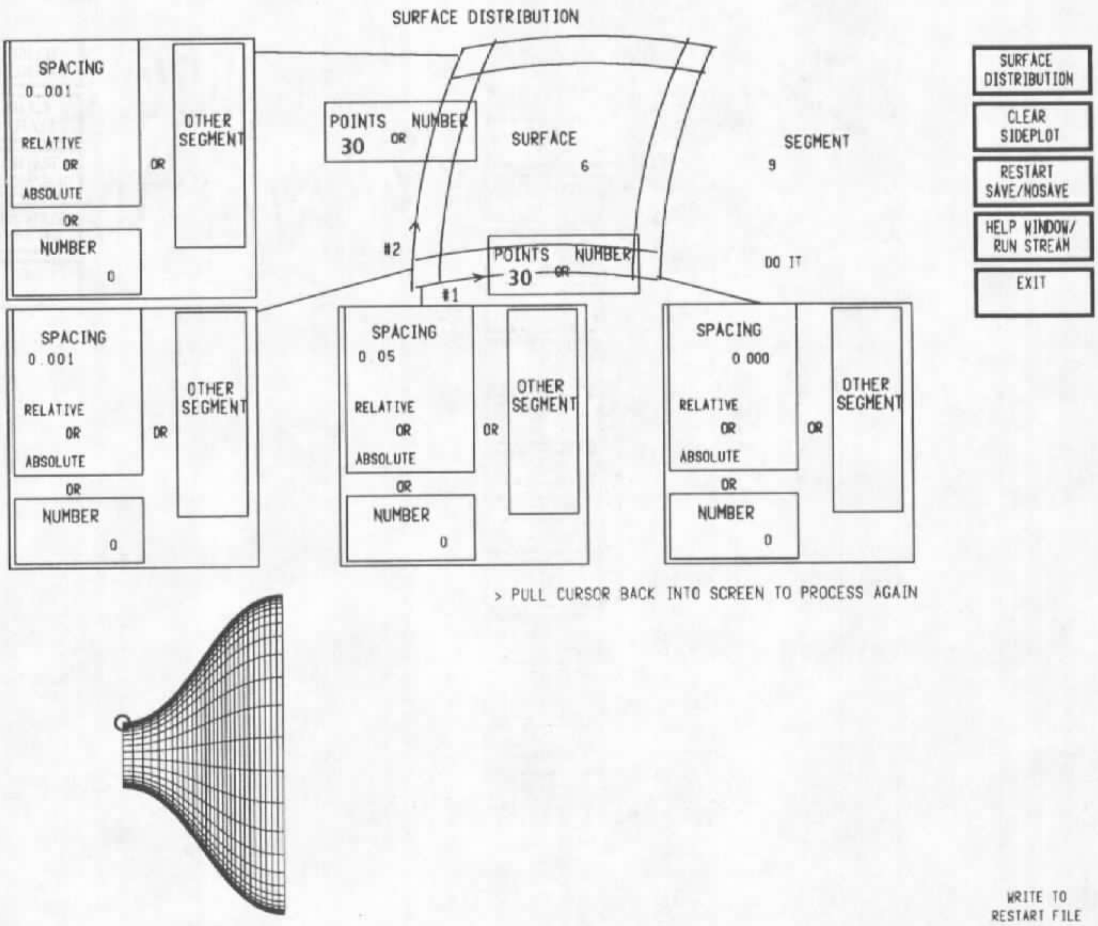


Figure 51. SURFACE DISTRIBUTION form.

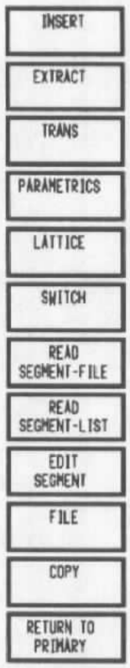


Figure 52. UTILITIES menu.

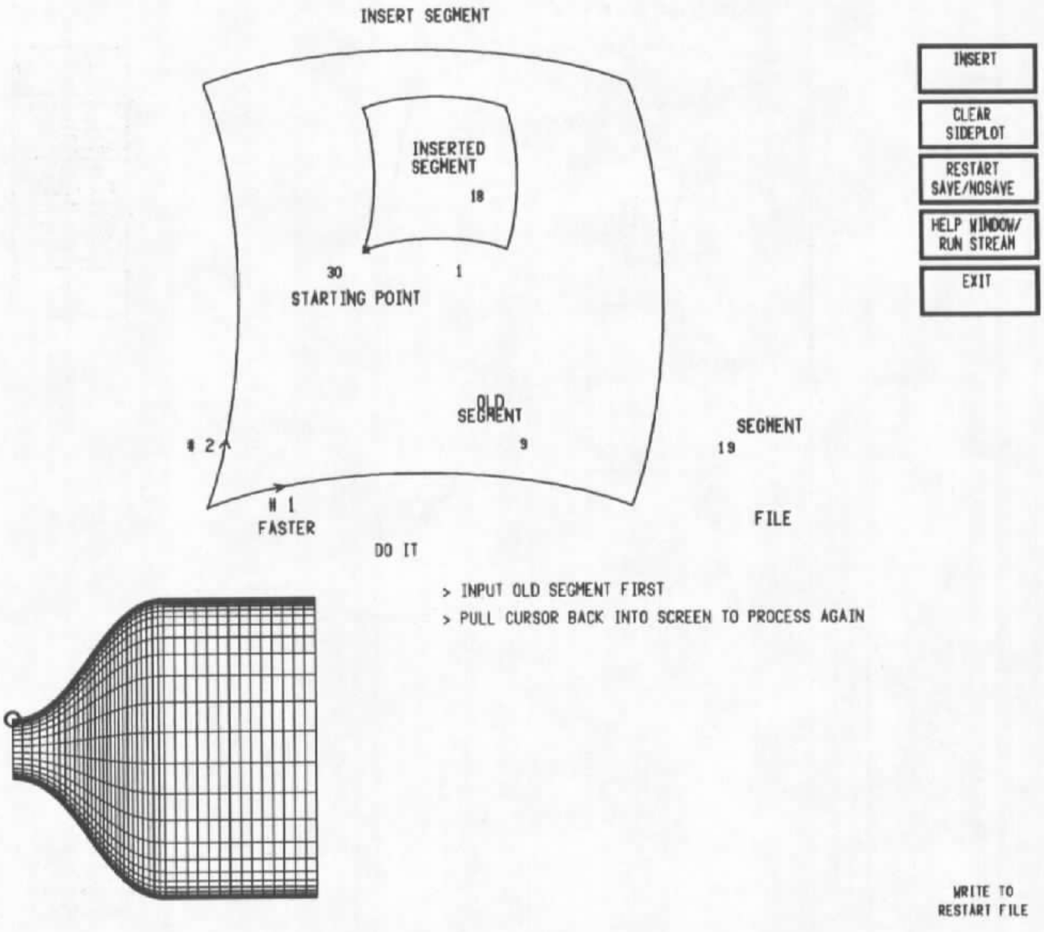


Figure 53. INSERT form.

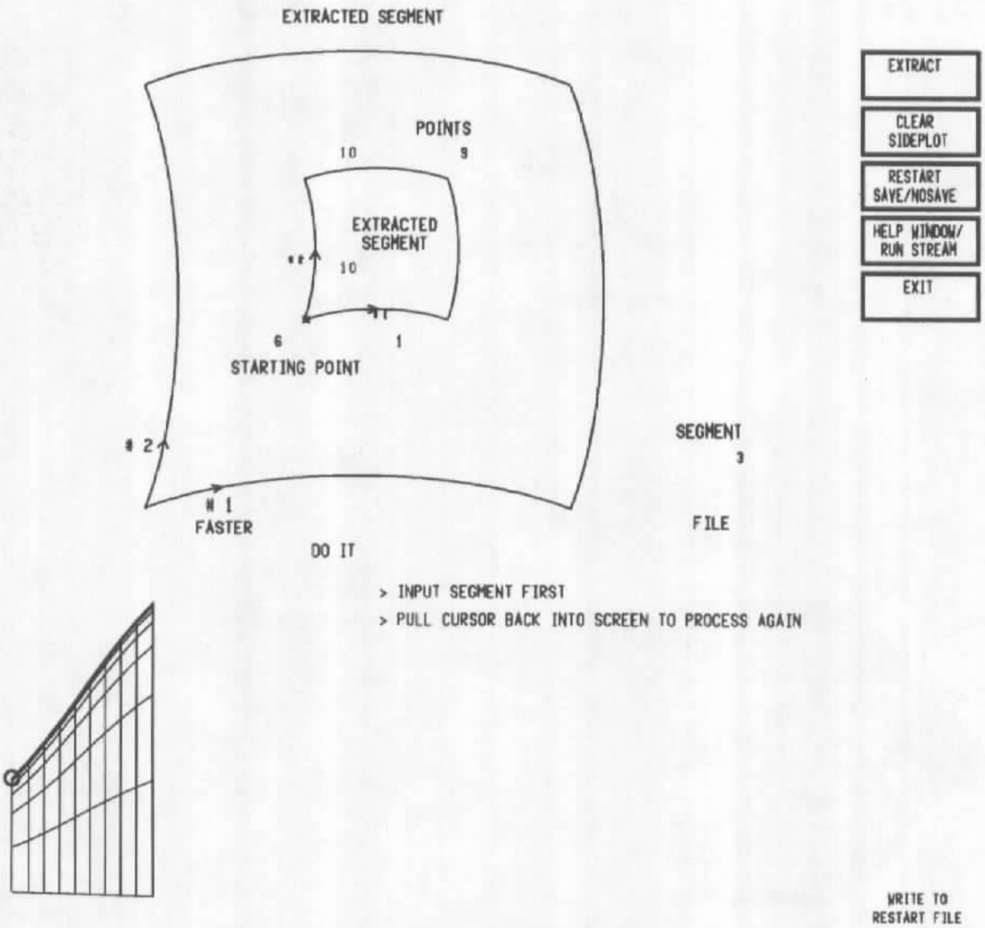


Figure 54. EXTRACT form.

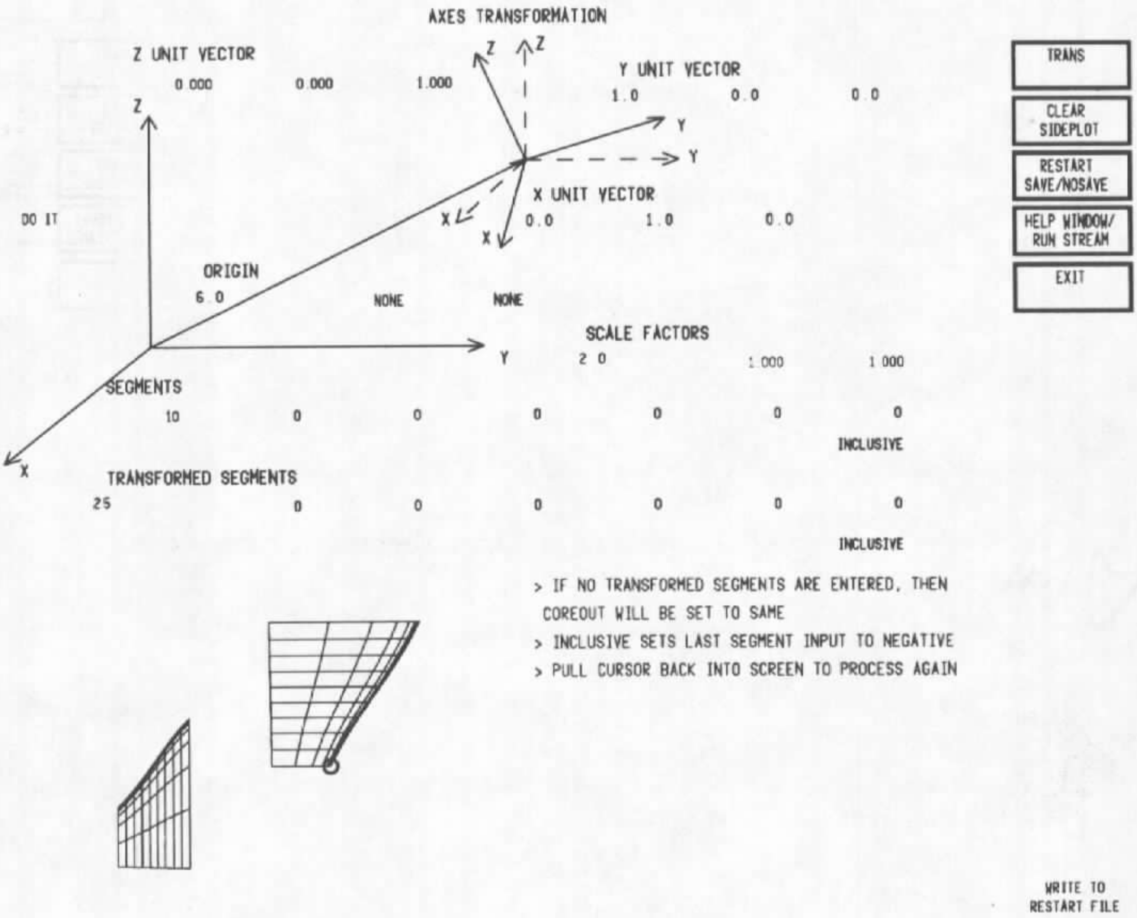


Figure 55. TRANS form.

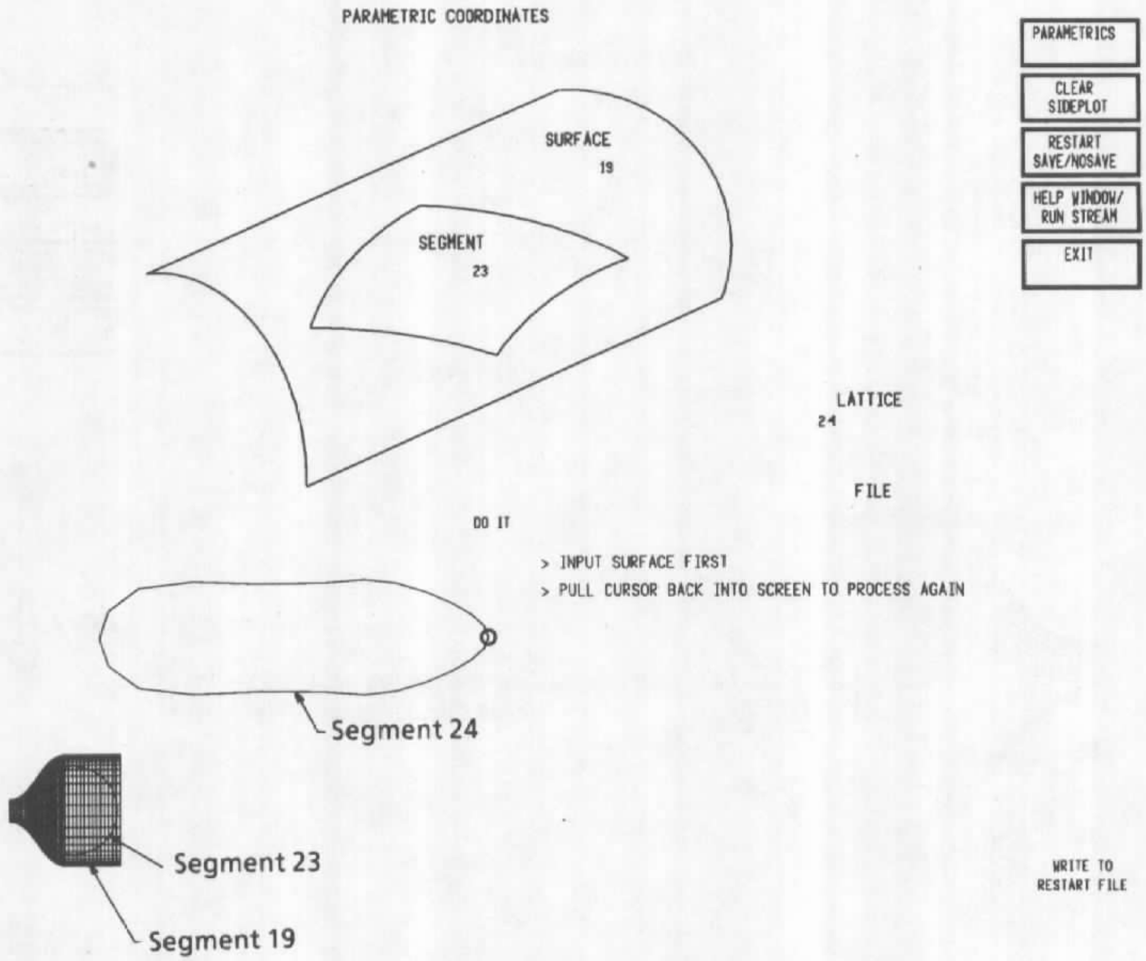


Figure 56. PARAMETRICS form.

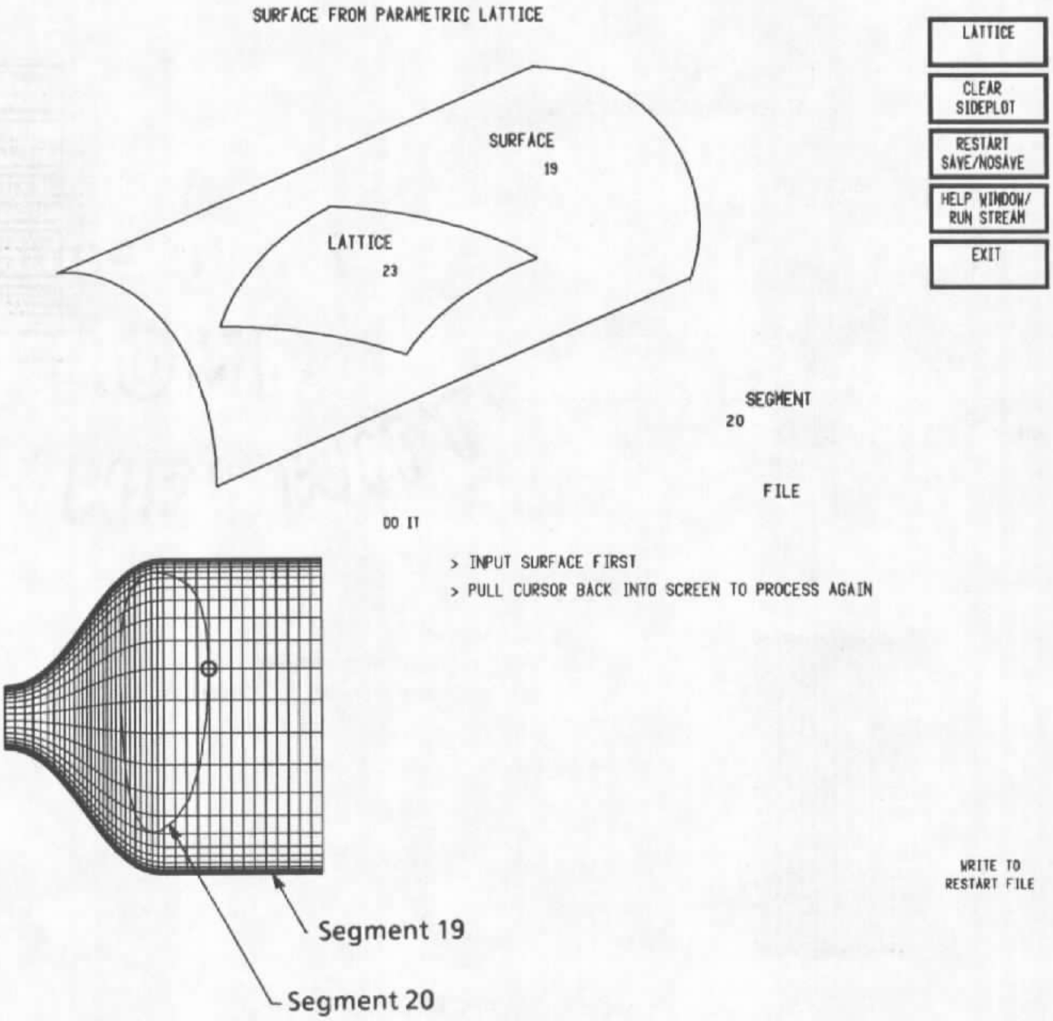


Figure 57. LATTICE form.

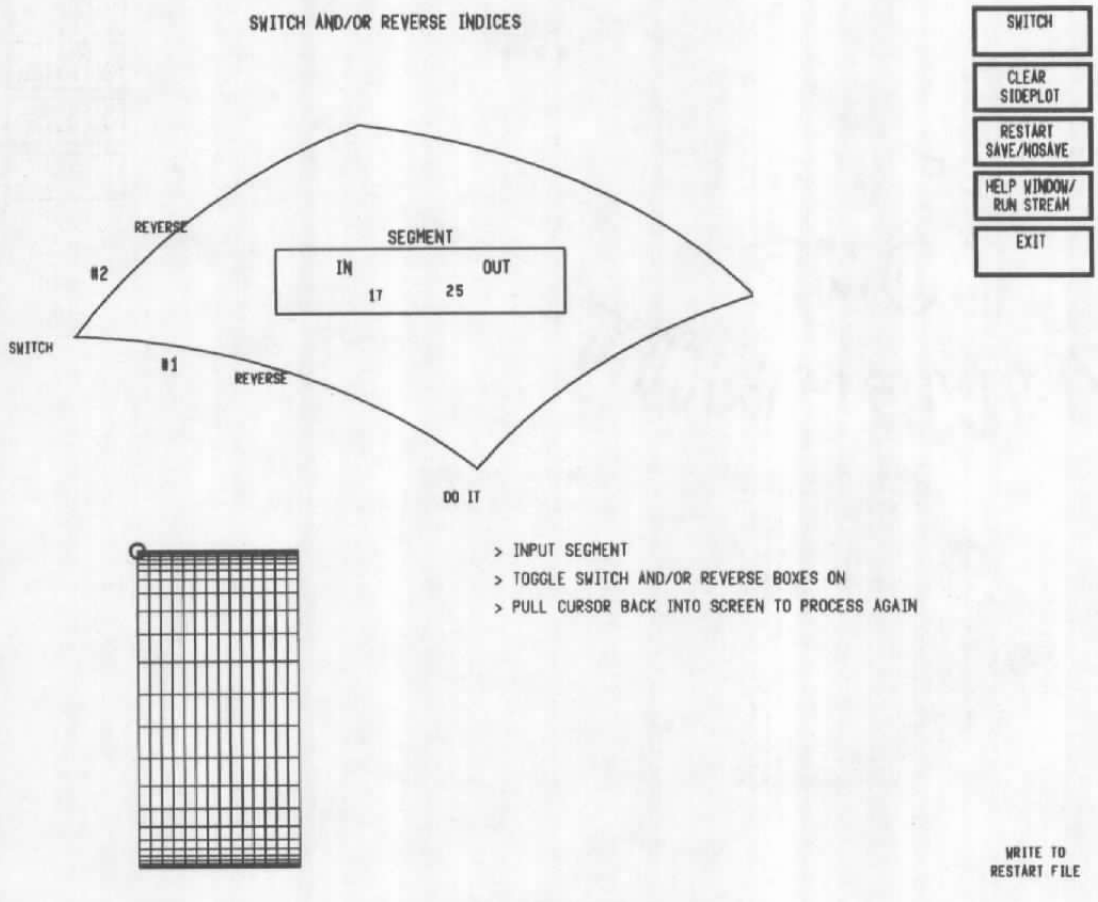


Figure 58. SWITCH form.

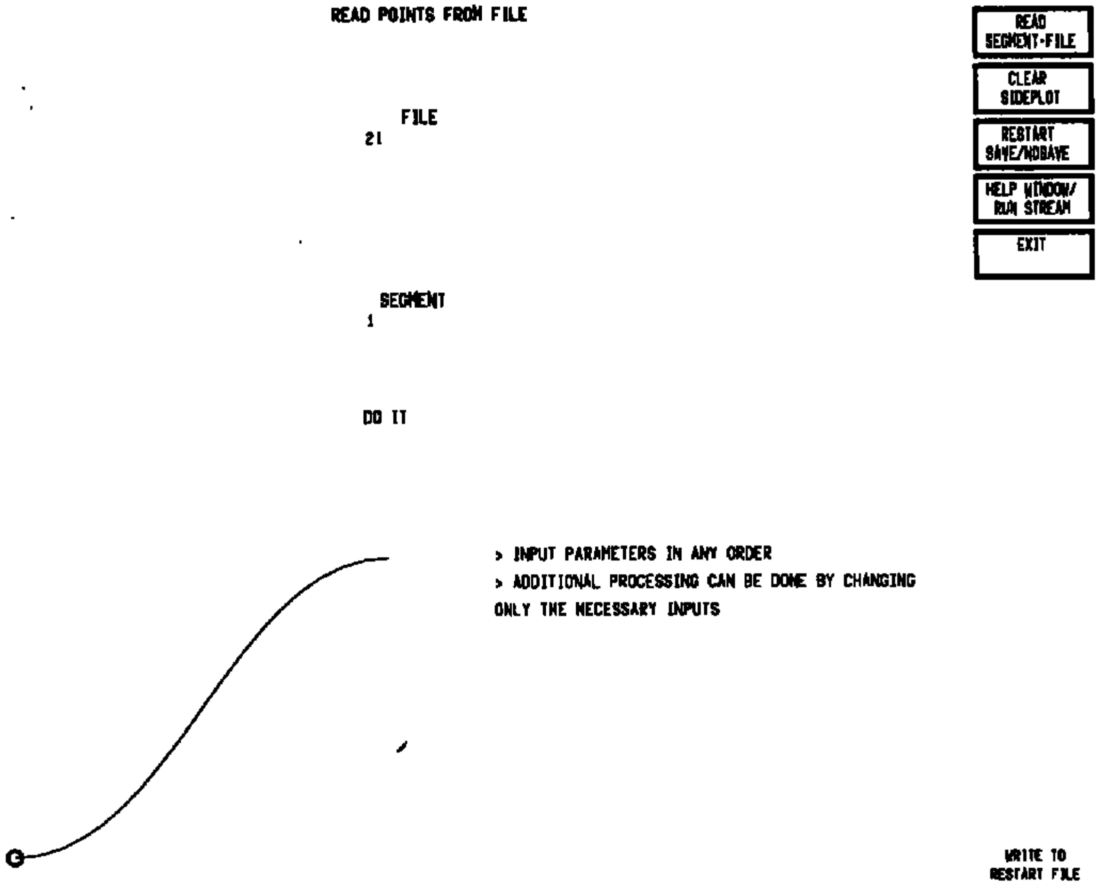


Figure 59. READ SEGMENT-FILE form.

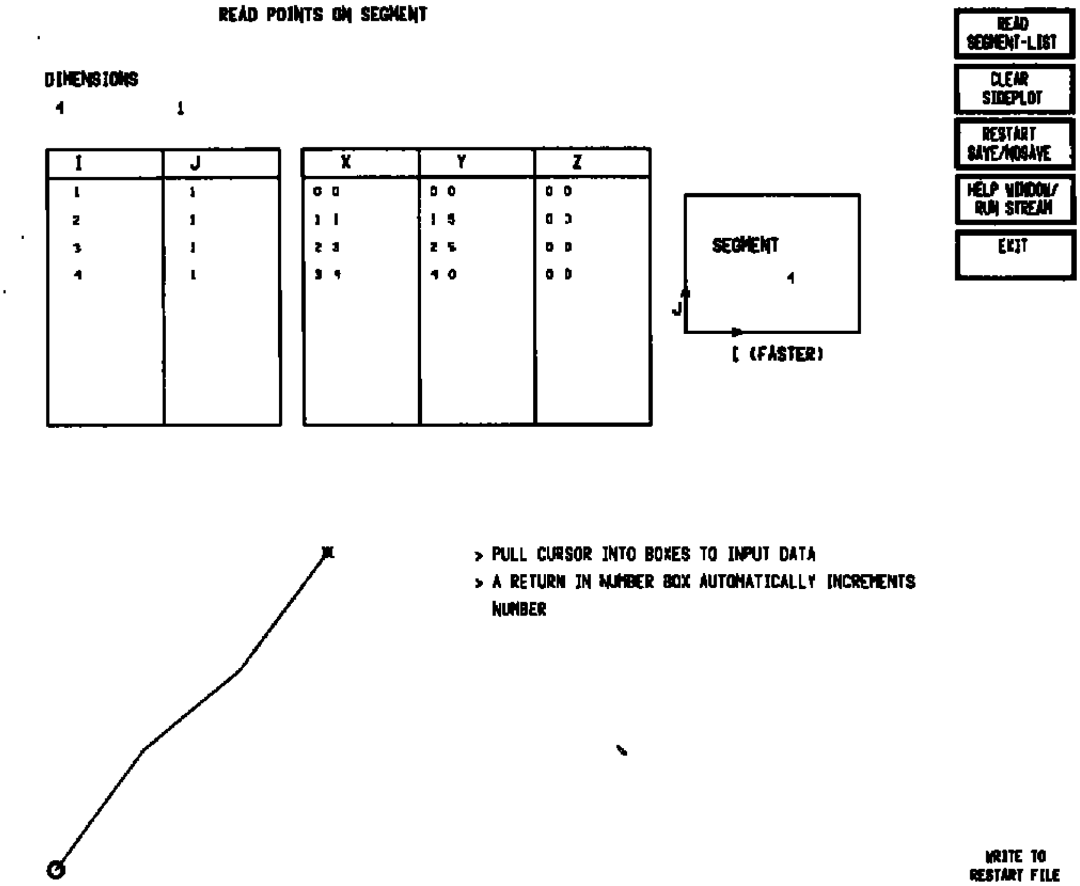


Figure 60. READ SEGMENT-LIST form.

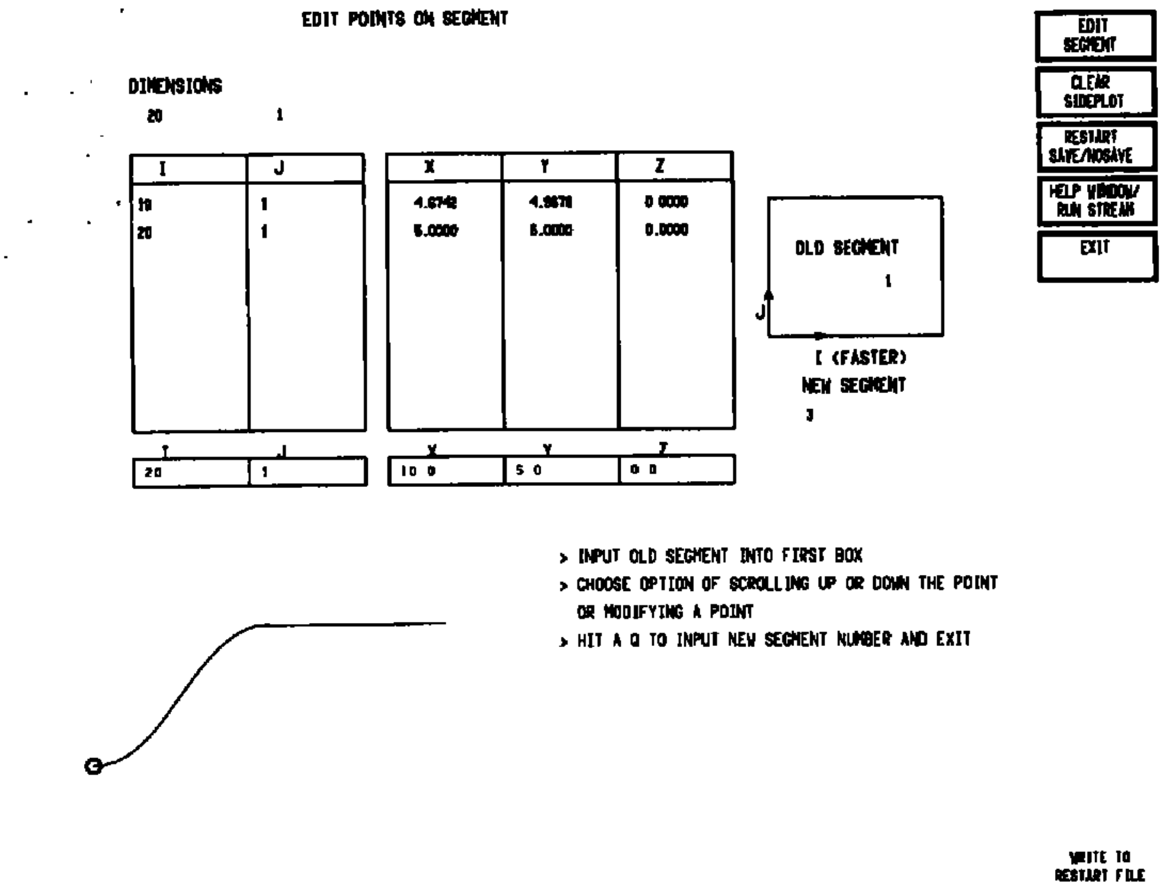


Figure 61. EDIT SEGMENT form.

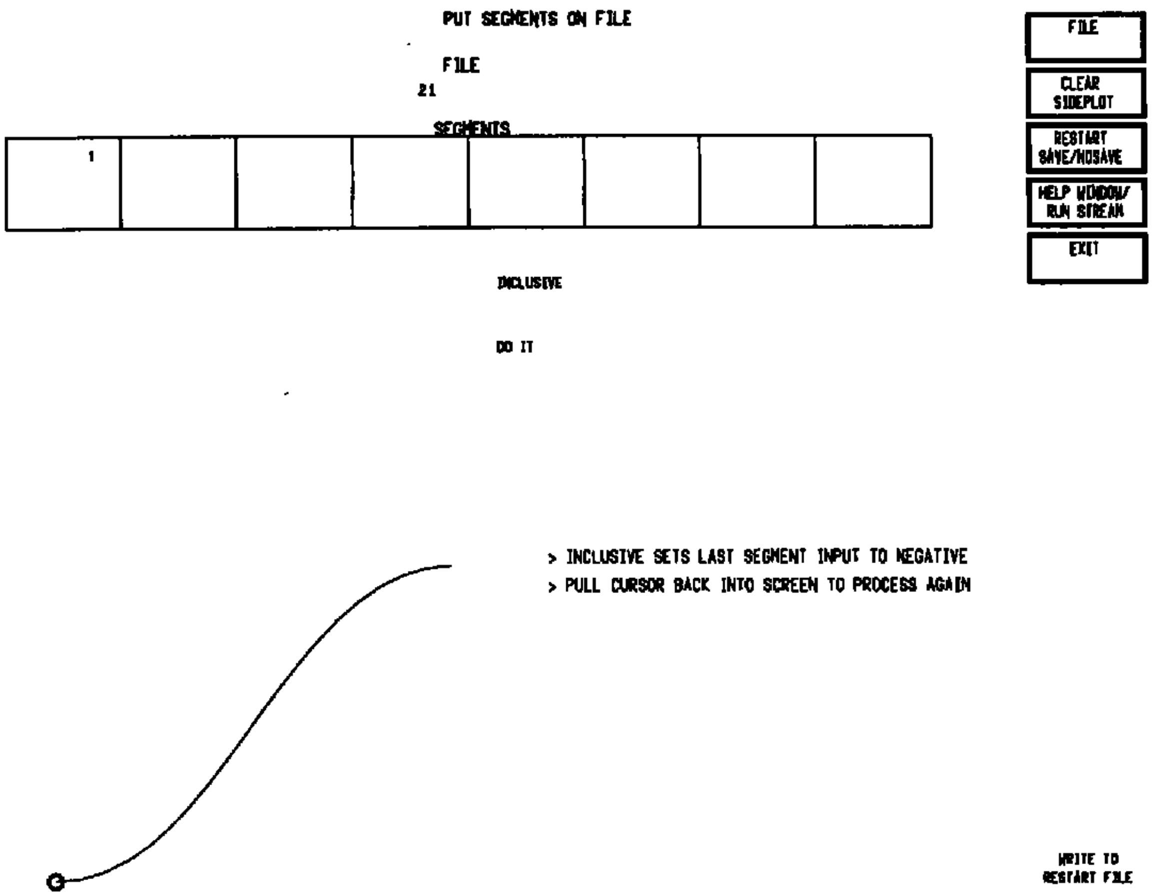


Figure 62. FILE form.

Figure 63. PROFILES menu.

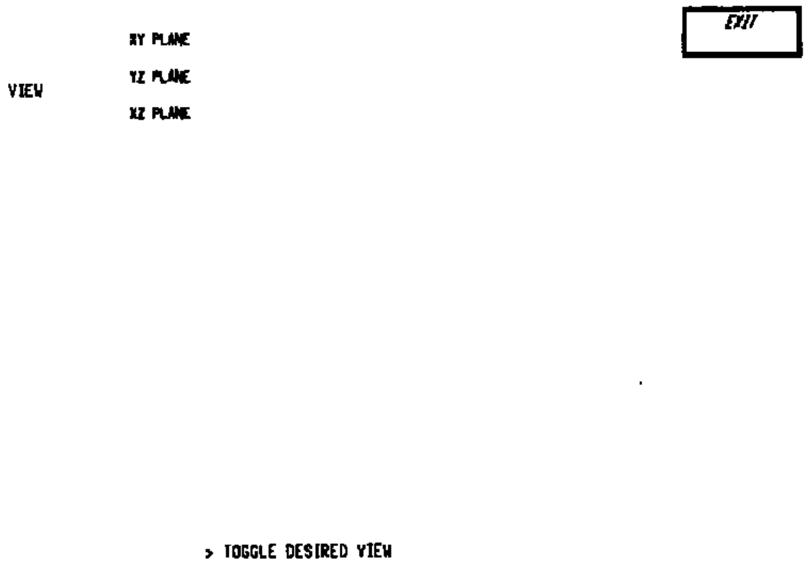


Figure 64. SELECT VIEW PLANE form.

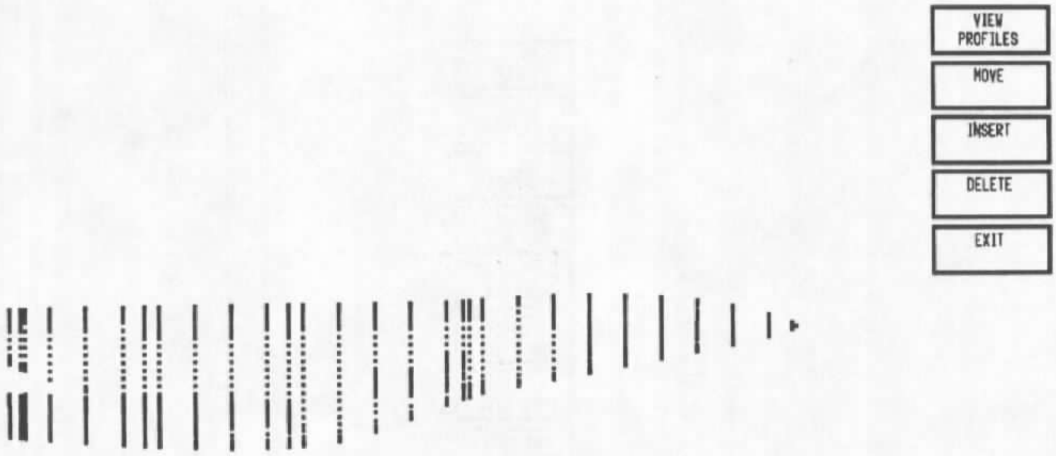


Figure 65. EDIT PROFILE(S) screen.

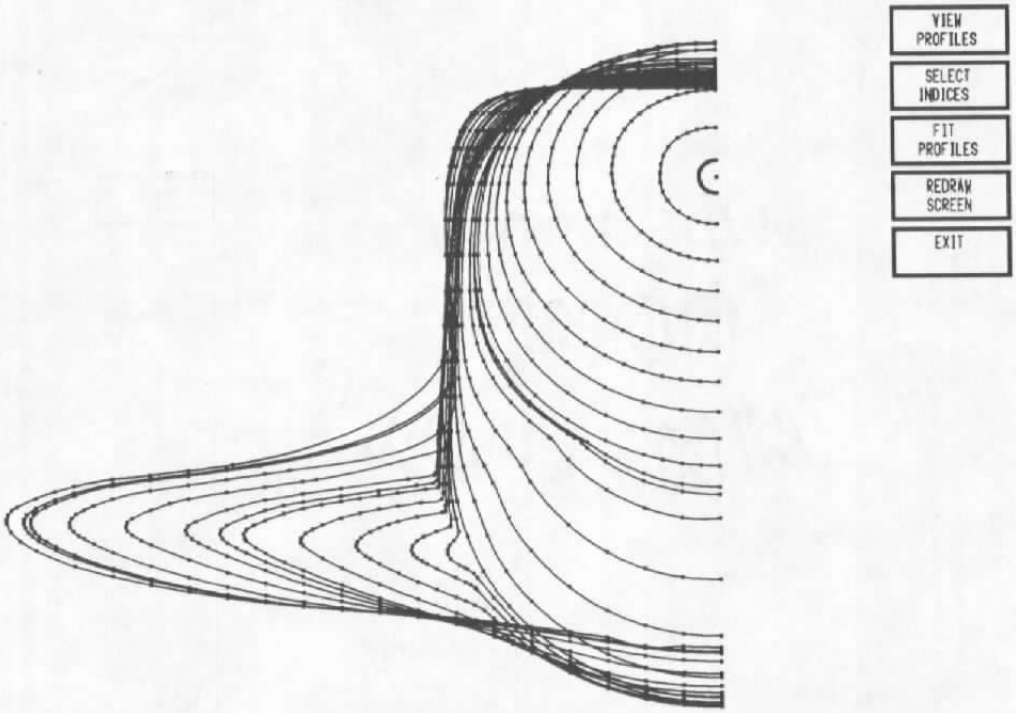


Figure 66. FIT AROUND PROFILE(S) screen.

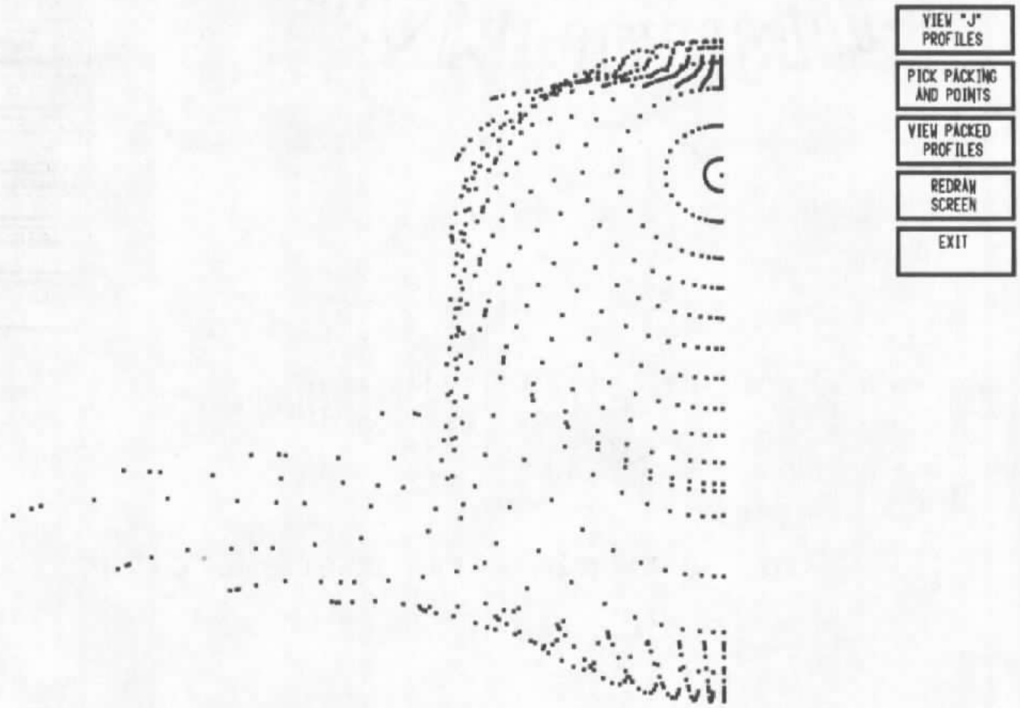


Figure 67. PACK AROUND PROFILE(S) screen.

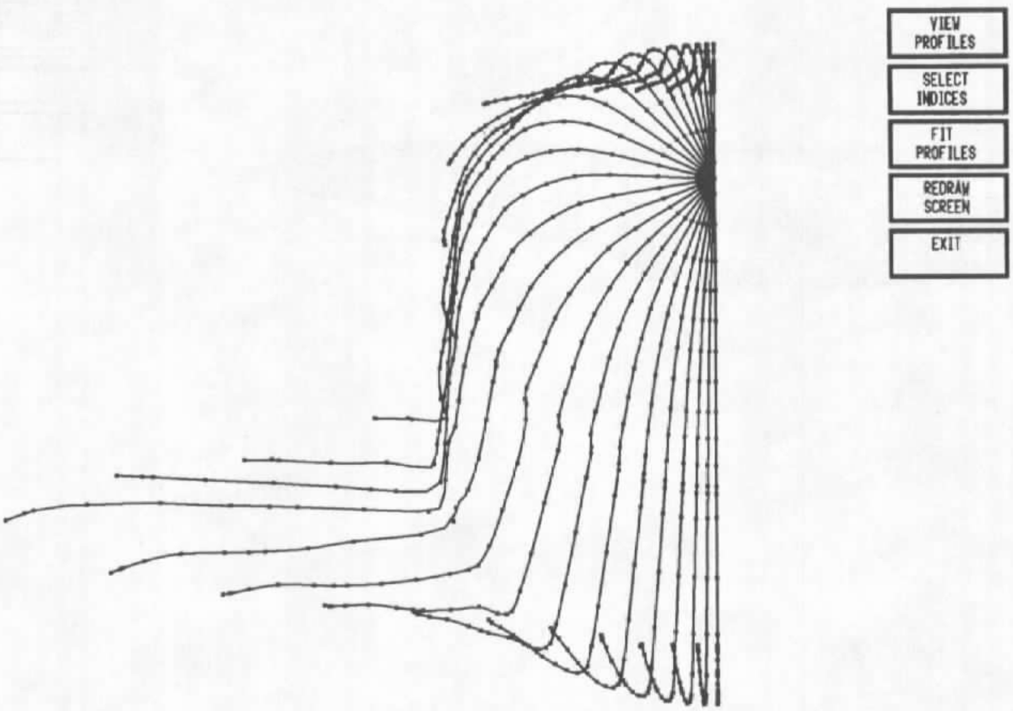


Figure 68. FIT DOWN PROFILE(S) screen.

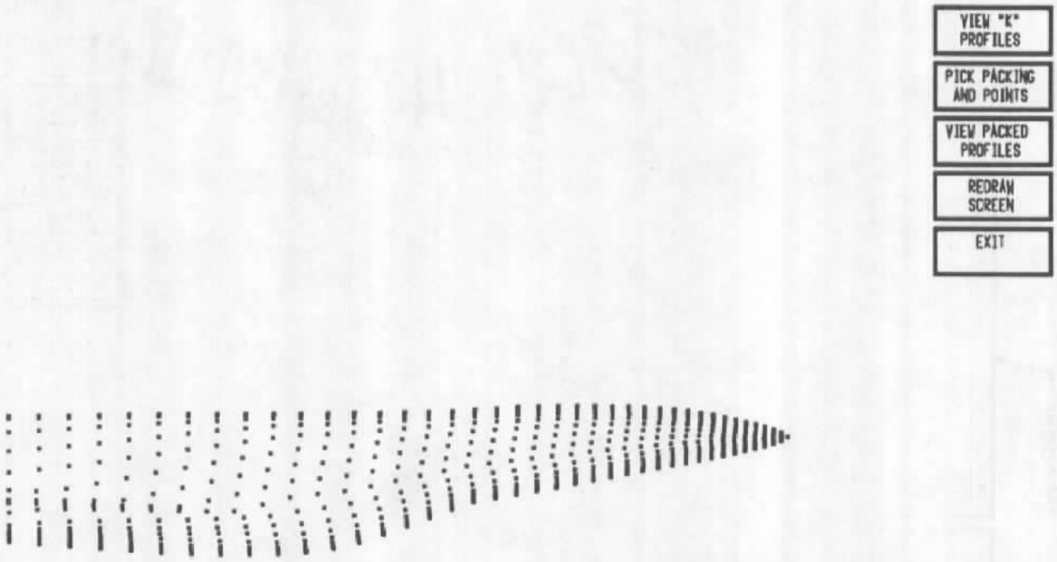


Figure 69. PACK DOWN PROFILE(S) screen.

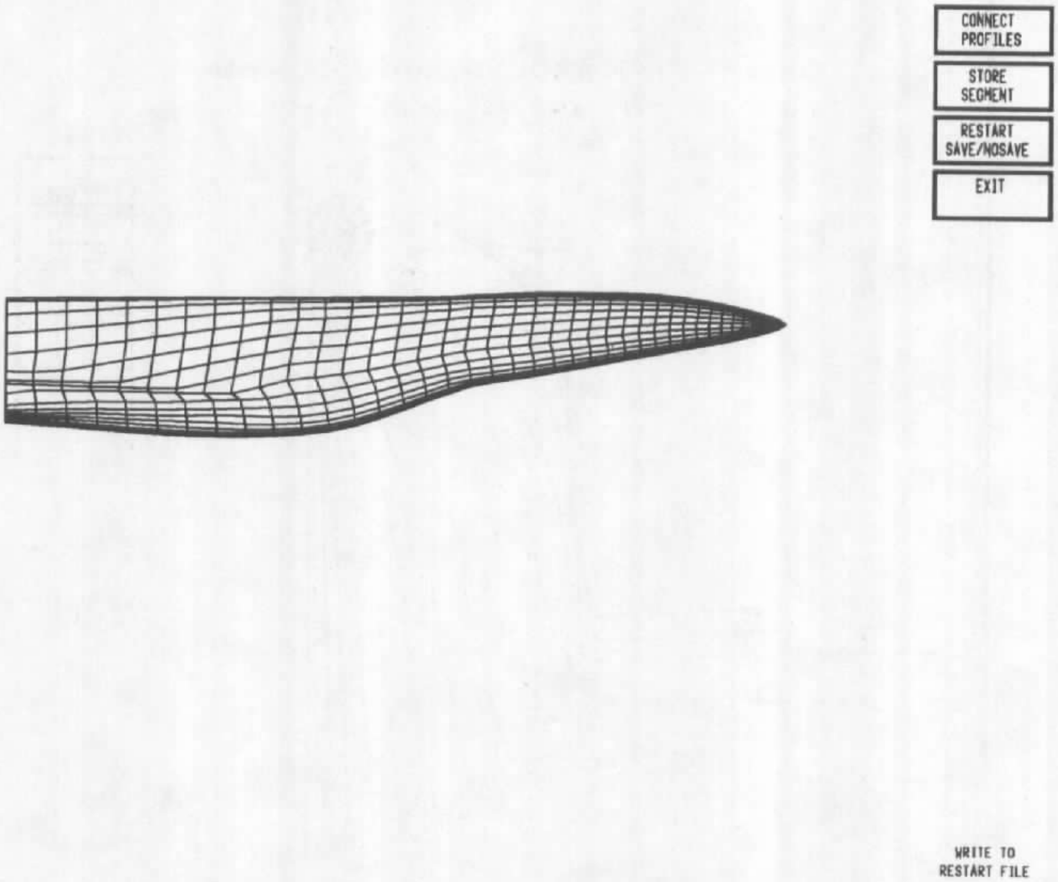


Figure 70. CONNECT PROFILE(S) screen.

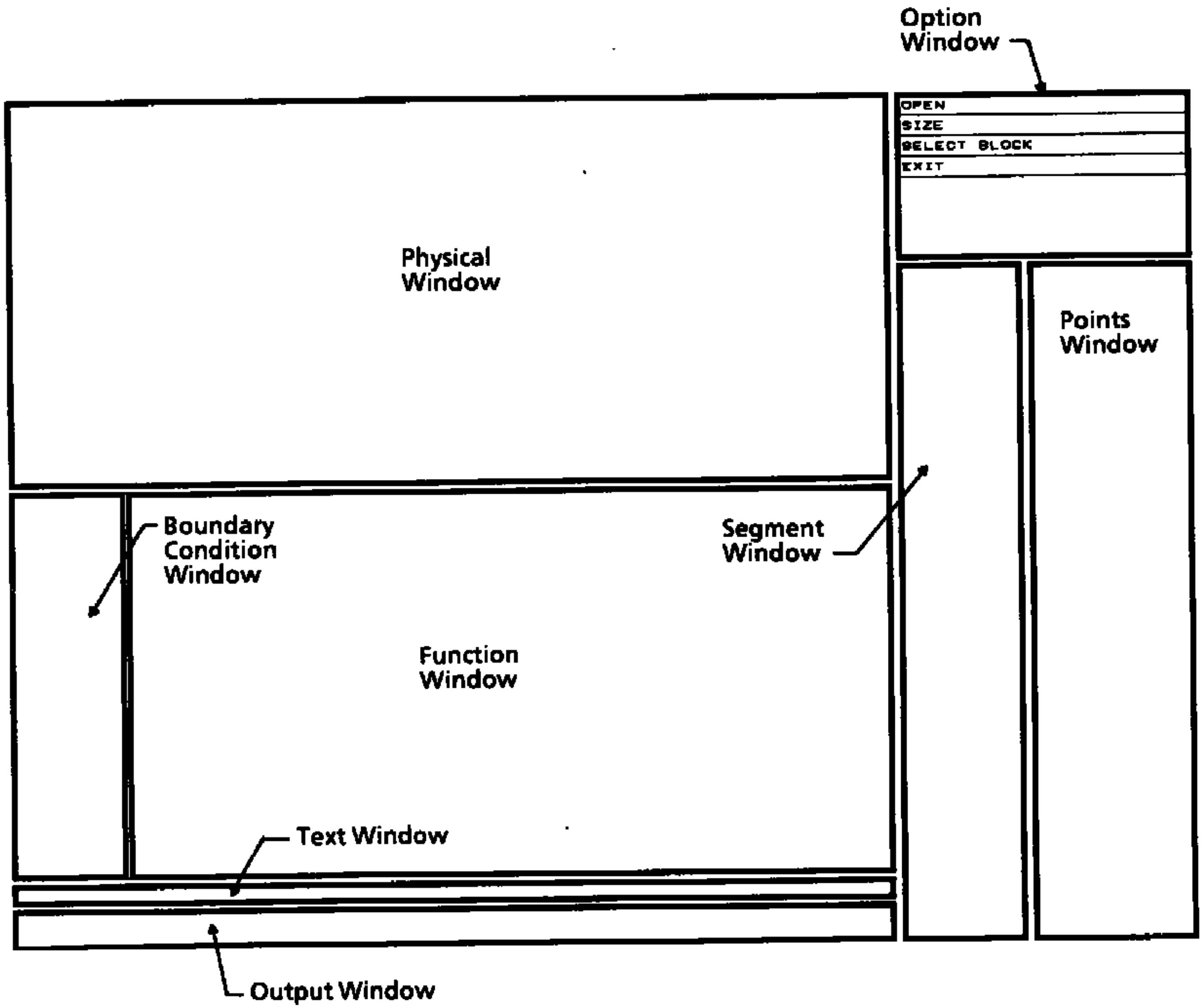


Figure 71. Grid generator—general screen layout and main menu.

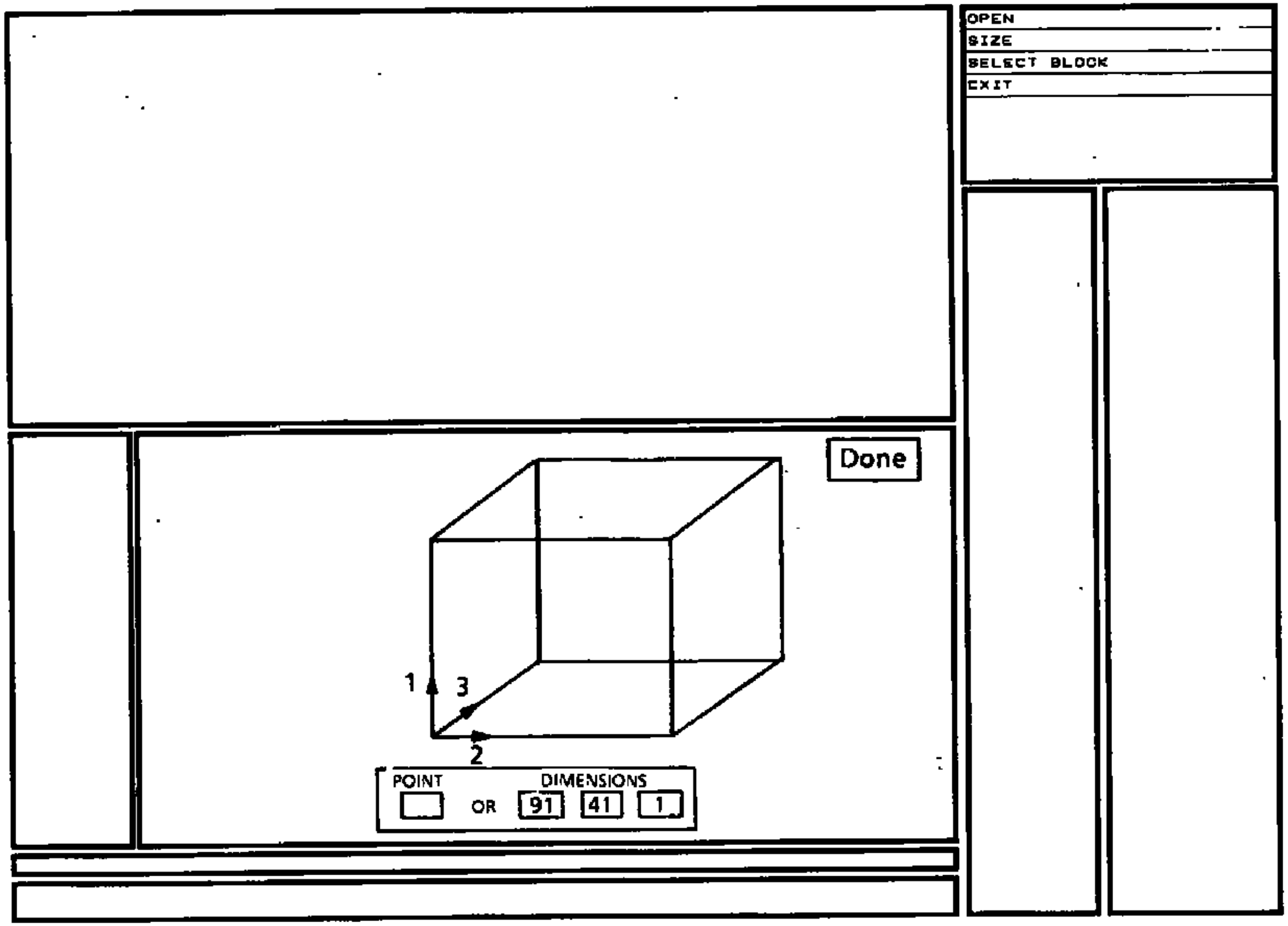


Figure 72. BLOCK option.

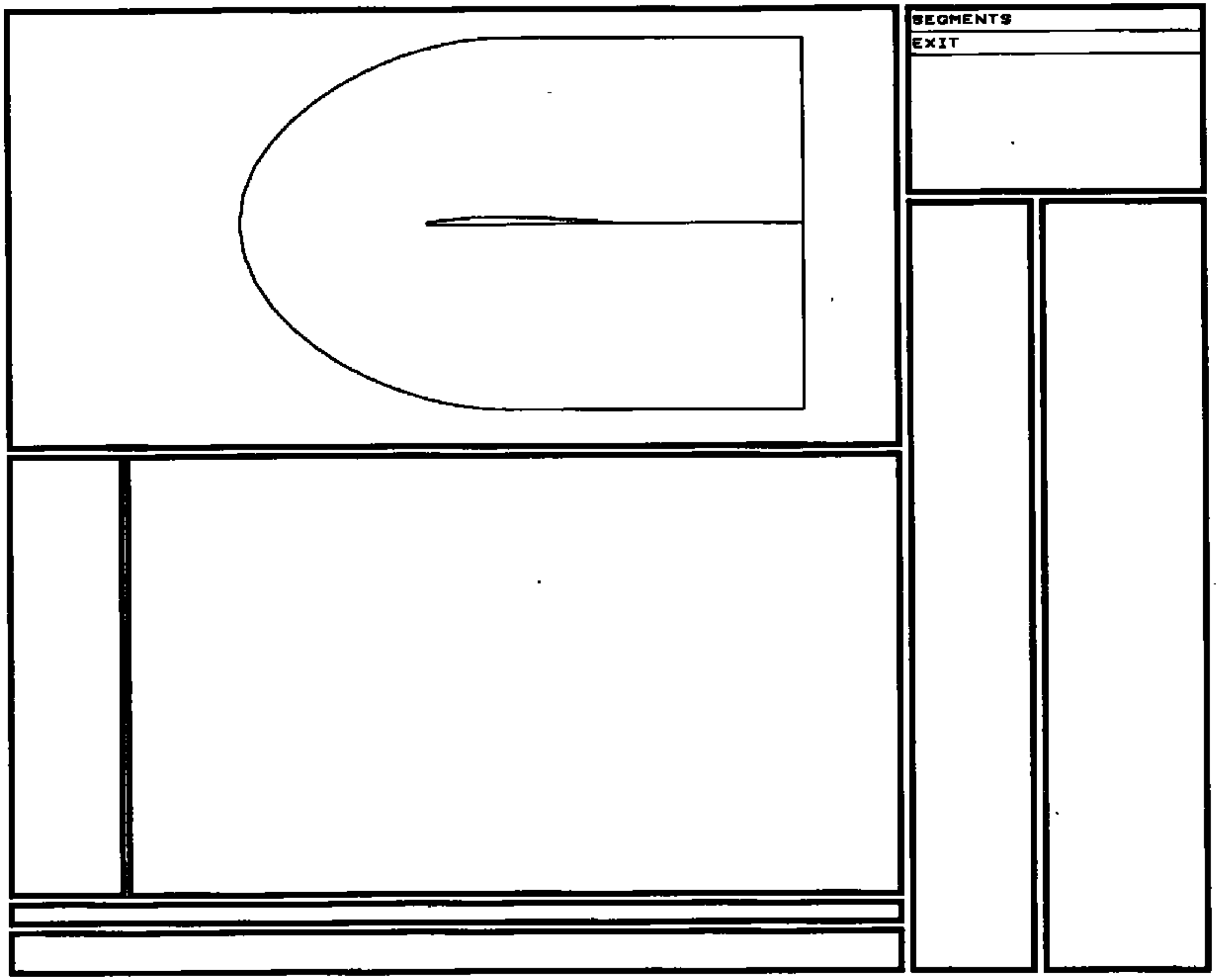


Figure 73. READ option.

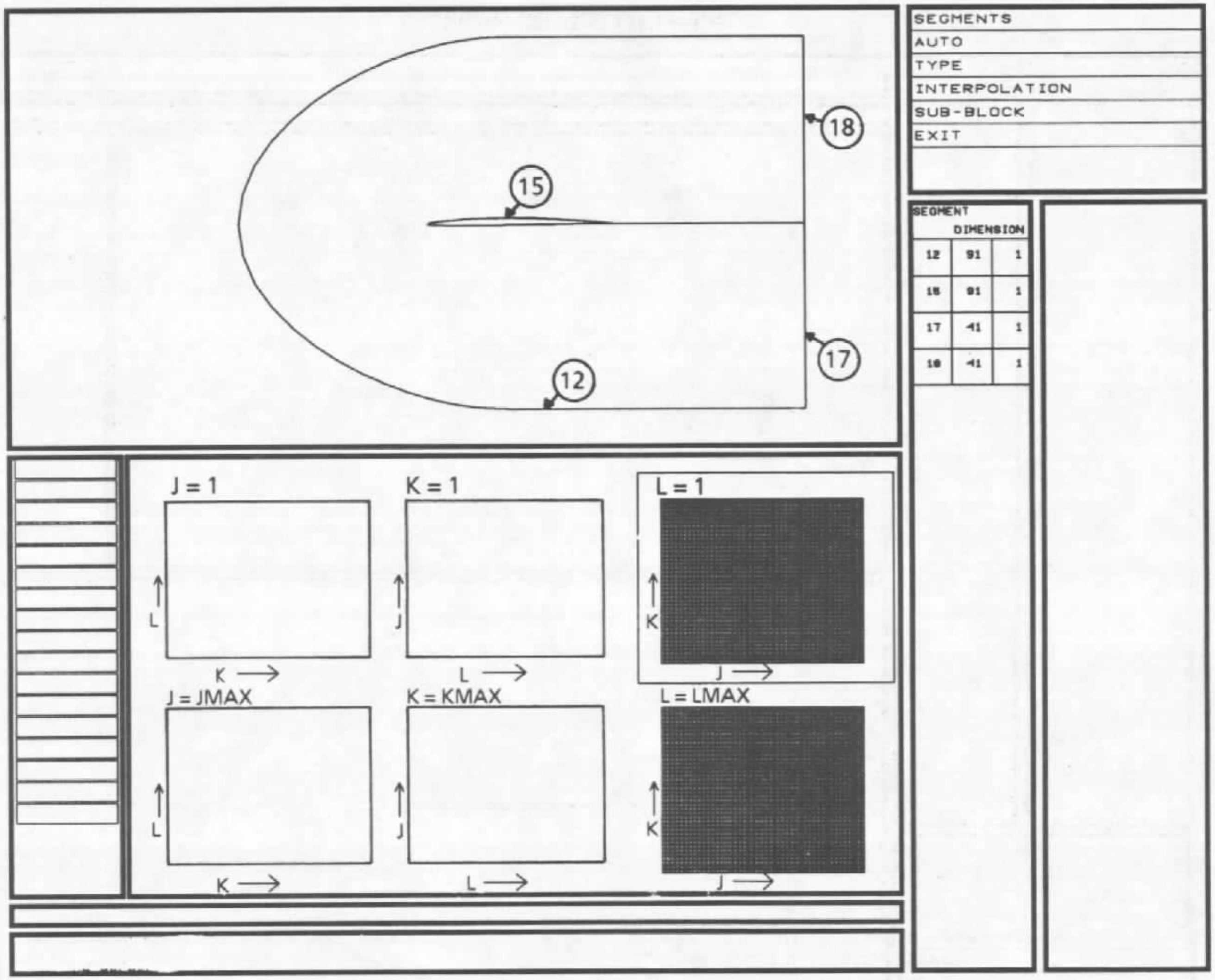


Figure 74. PLACE option.

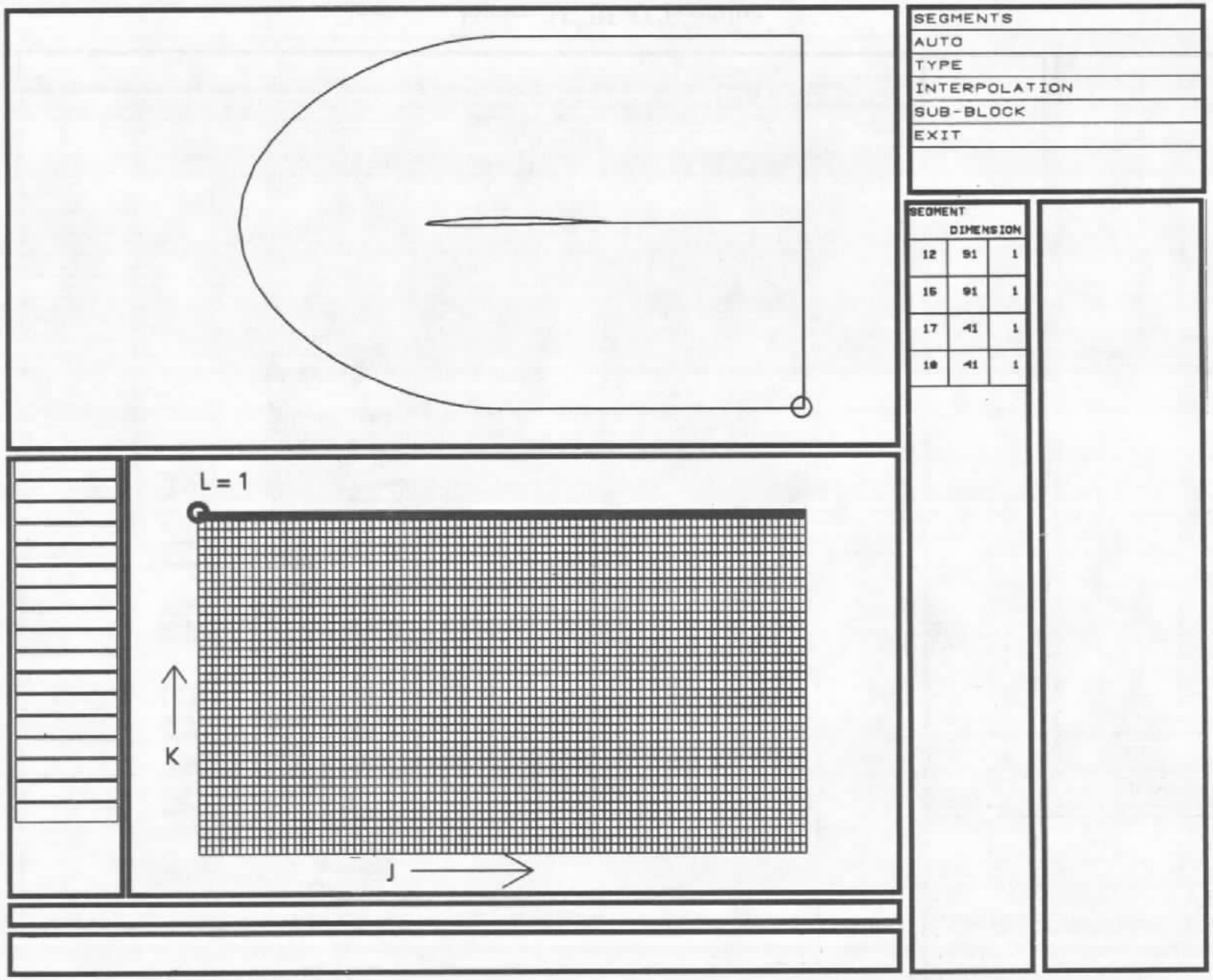


Figure 75. Placement of segments.

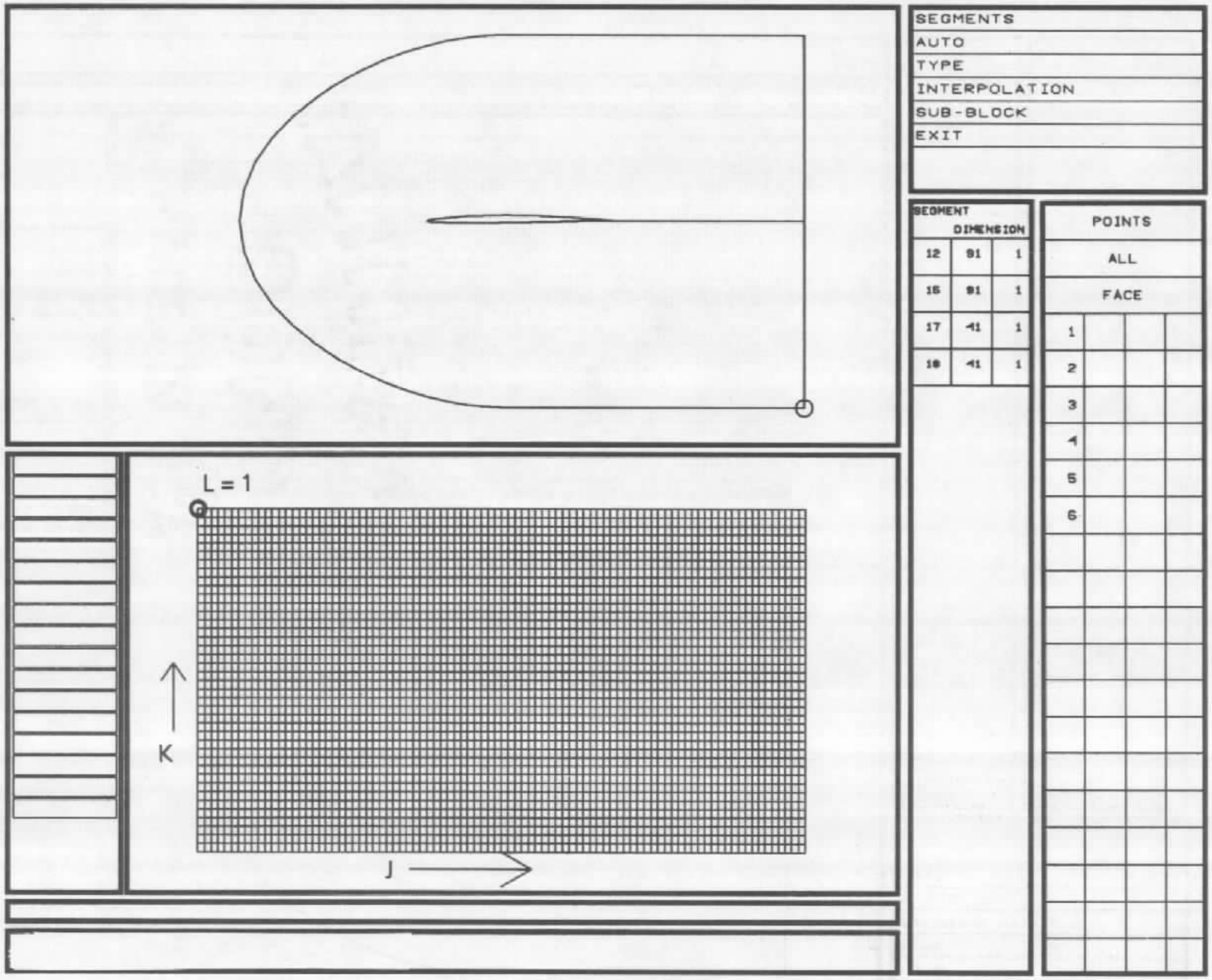


Figure 76. Selection of surfaces.

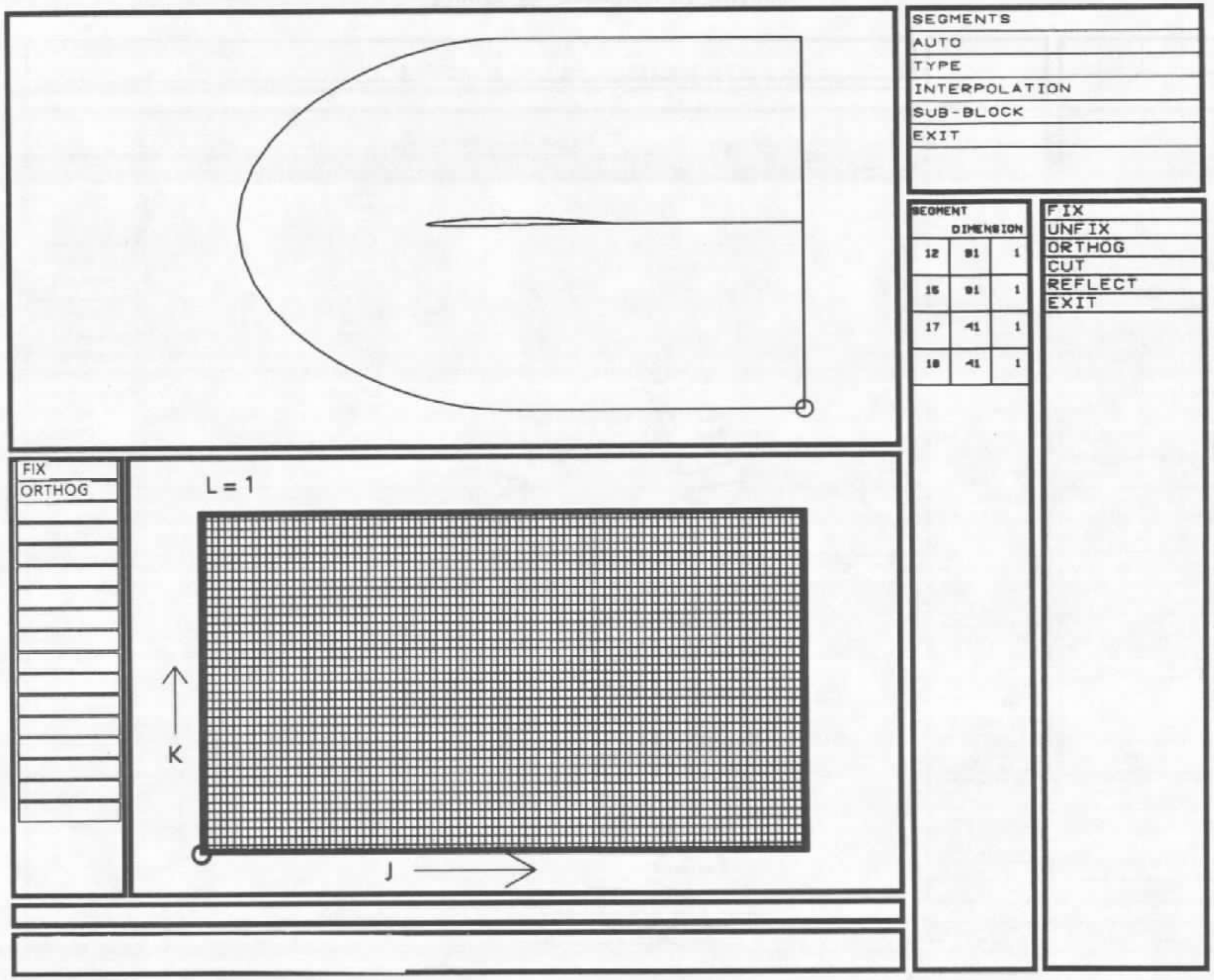


Figure 77. Placement of boundary conditions.

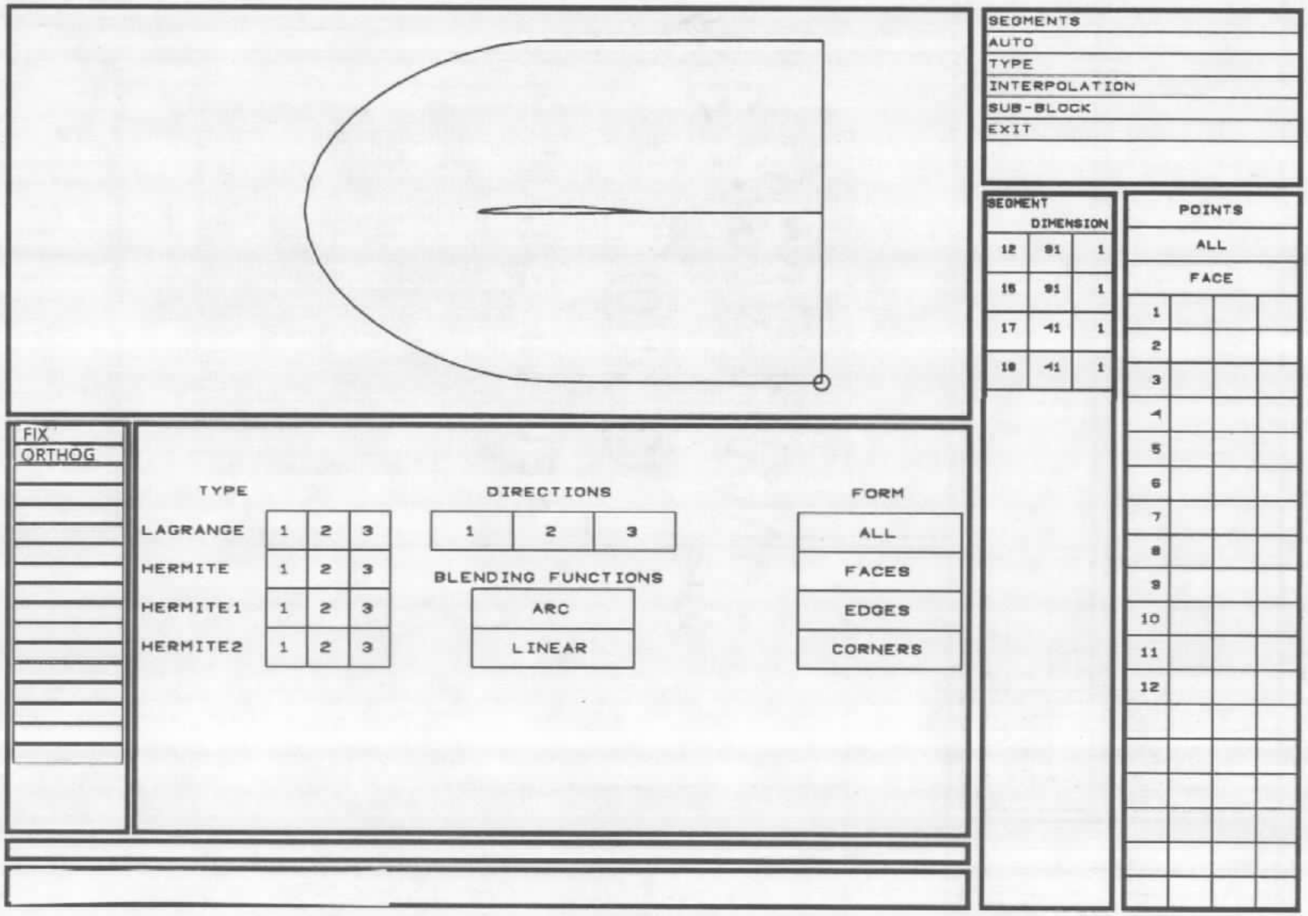


Figure 78. INTERPOLATION option.

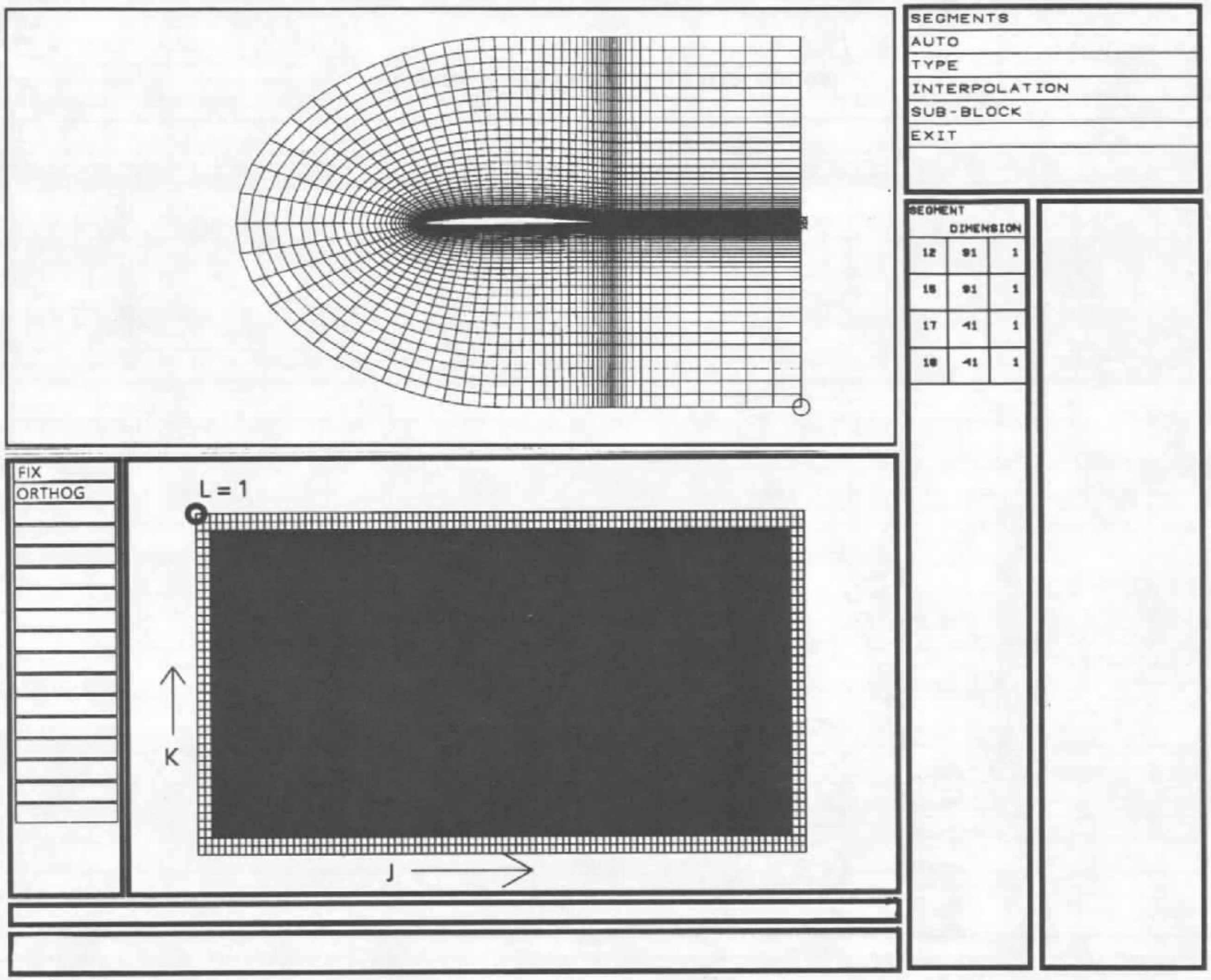


Figure 79. Generation of algebraic grid.

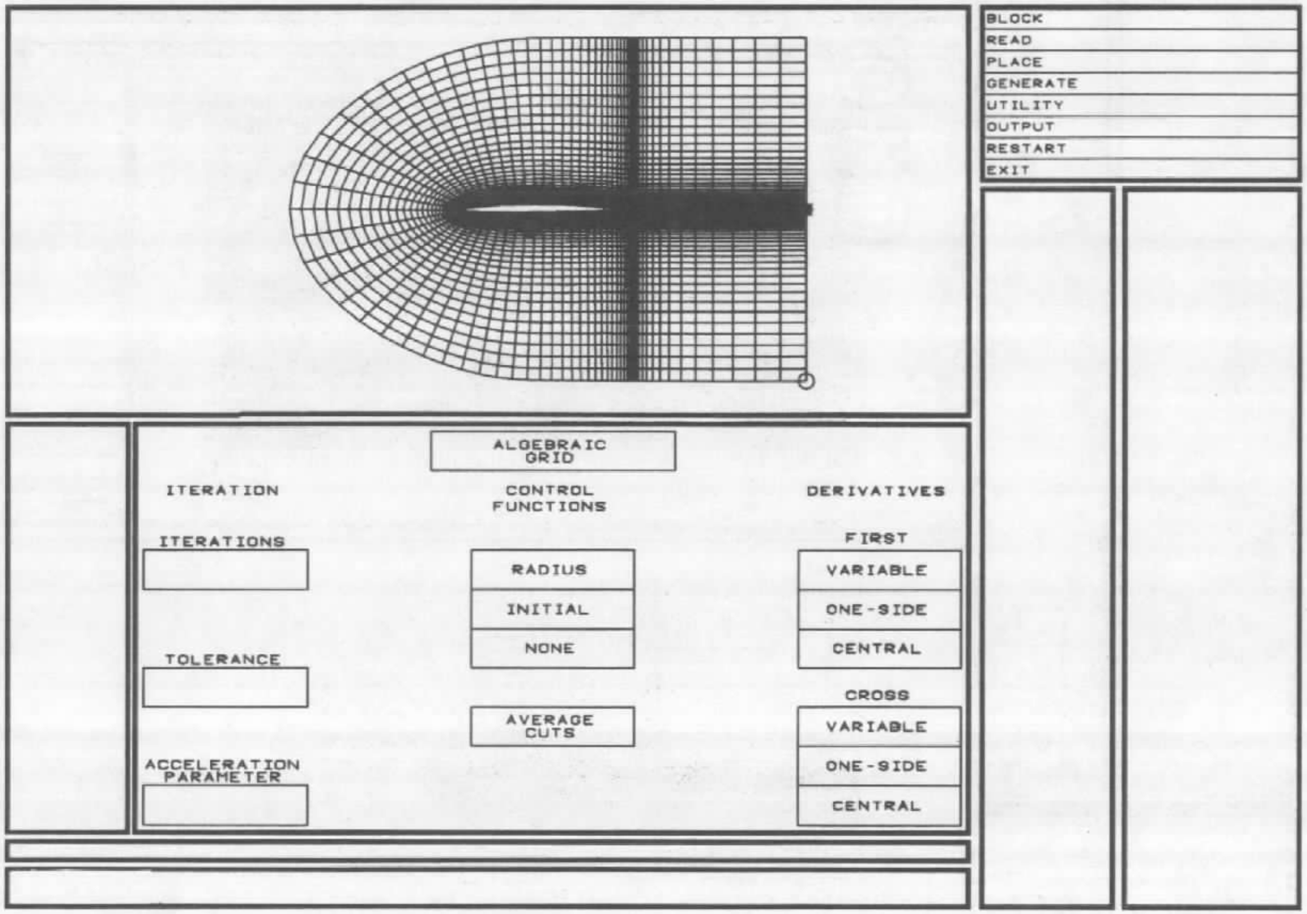


Figure 80. UTILITY option.

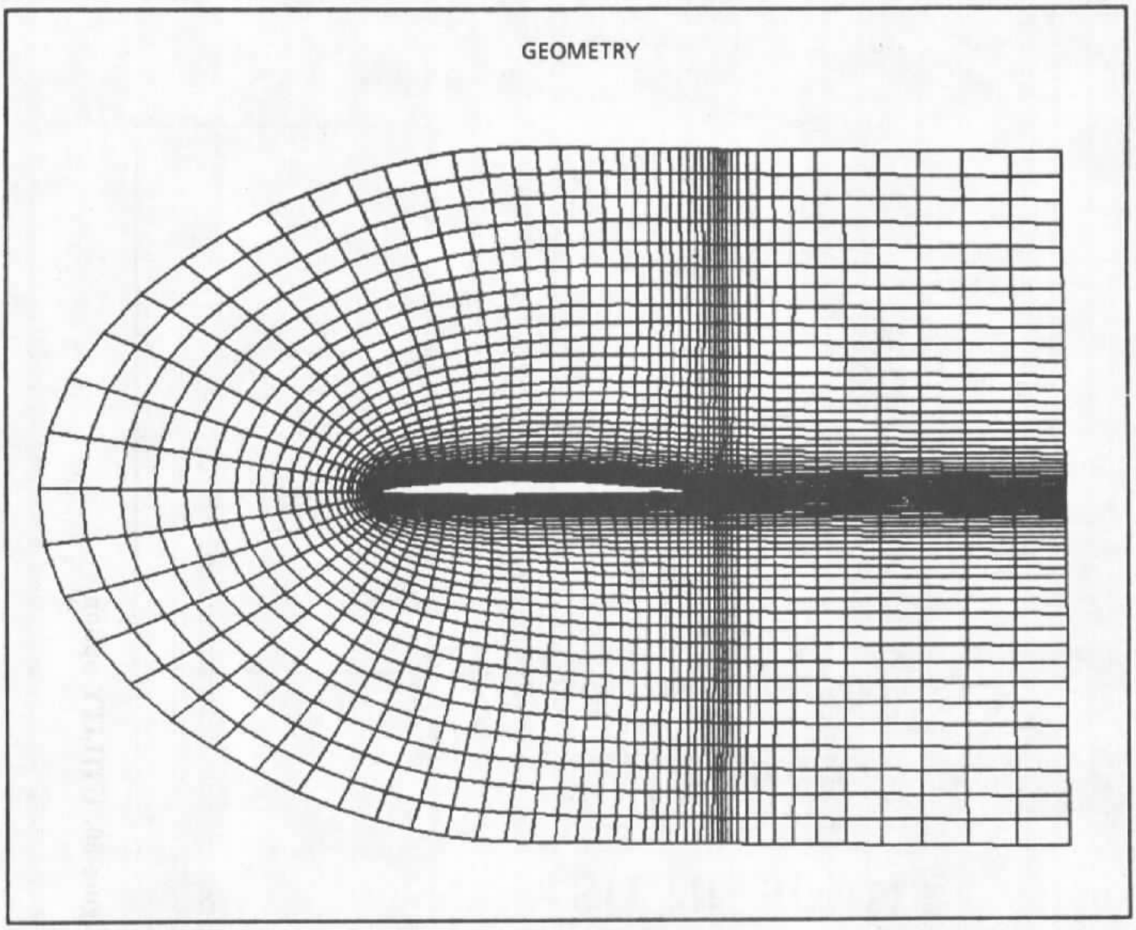


Figure 81. Final grid.

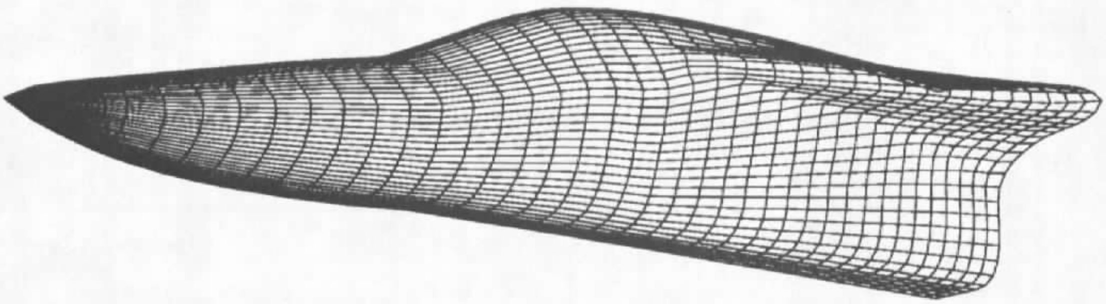


Figure 82. Body surface generated from profiles.

LABEL	X	Y	Z
1	0	0	0
2	−0.33	0	0
3	0.4867	0	0.0706
4	0.4867	0	−0.0706
.	.	.	.
.	.	.	.
.	0	0	0

a. Point table

LABEL	SPACING
1	0.0012
2	0.035
3	0.025
4	0.038
.	.
.	.
.	.

b. Table of spacings

LABEL	NUMBER OF POINTS
1	40
2	41
3	31
4	52
.	.
.	.
.	.

c. Table of numbers of points

Figure 83. Tables of point coordinates, spacings, and numbers of points.

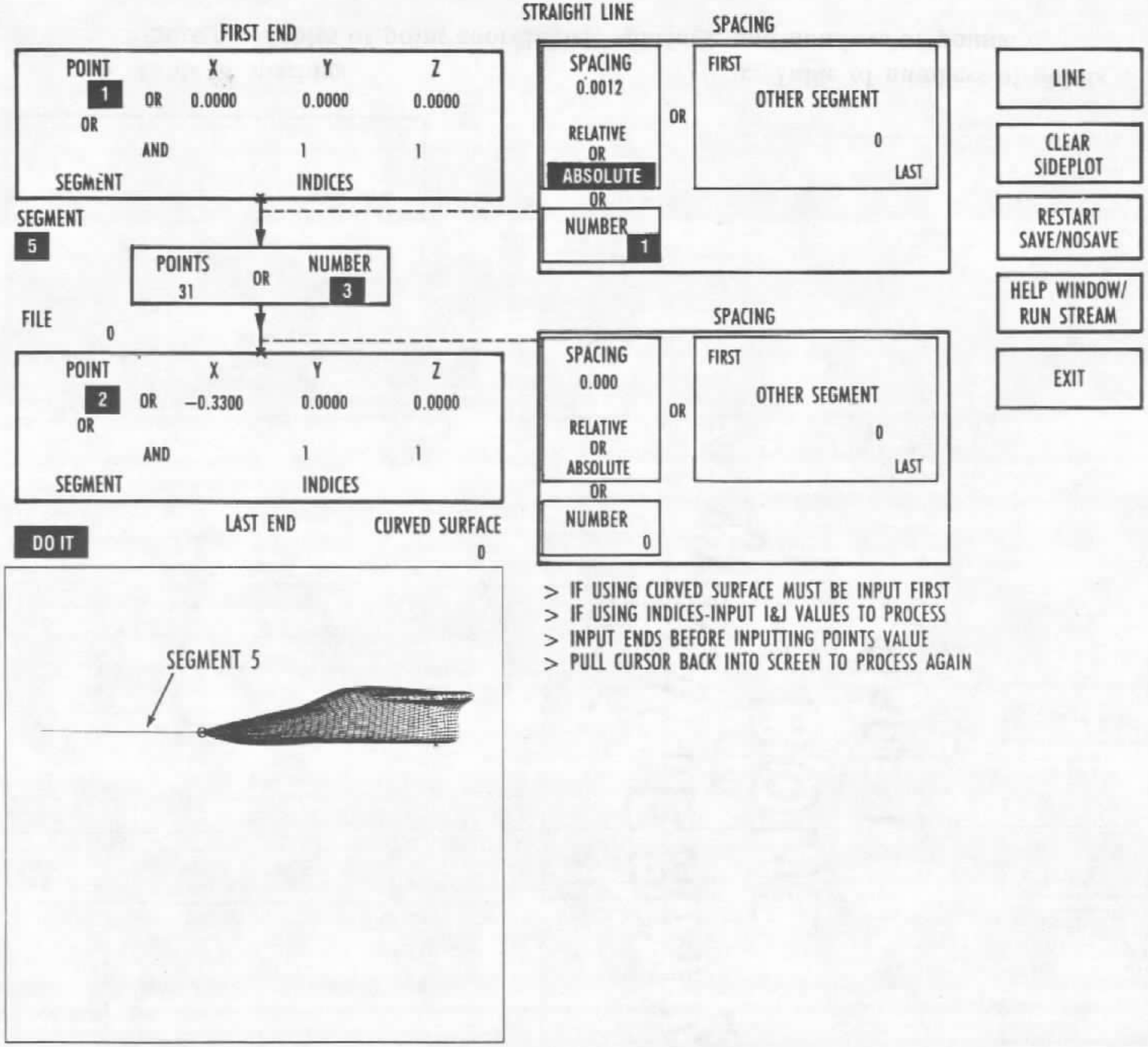


Figure 84. Generation of polar axis.

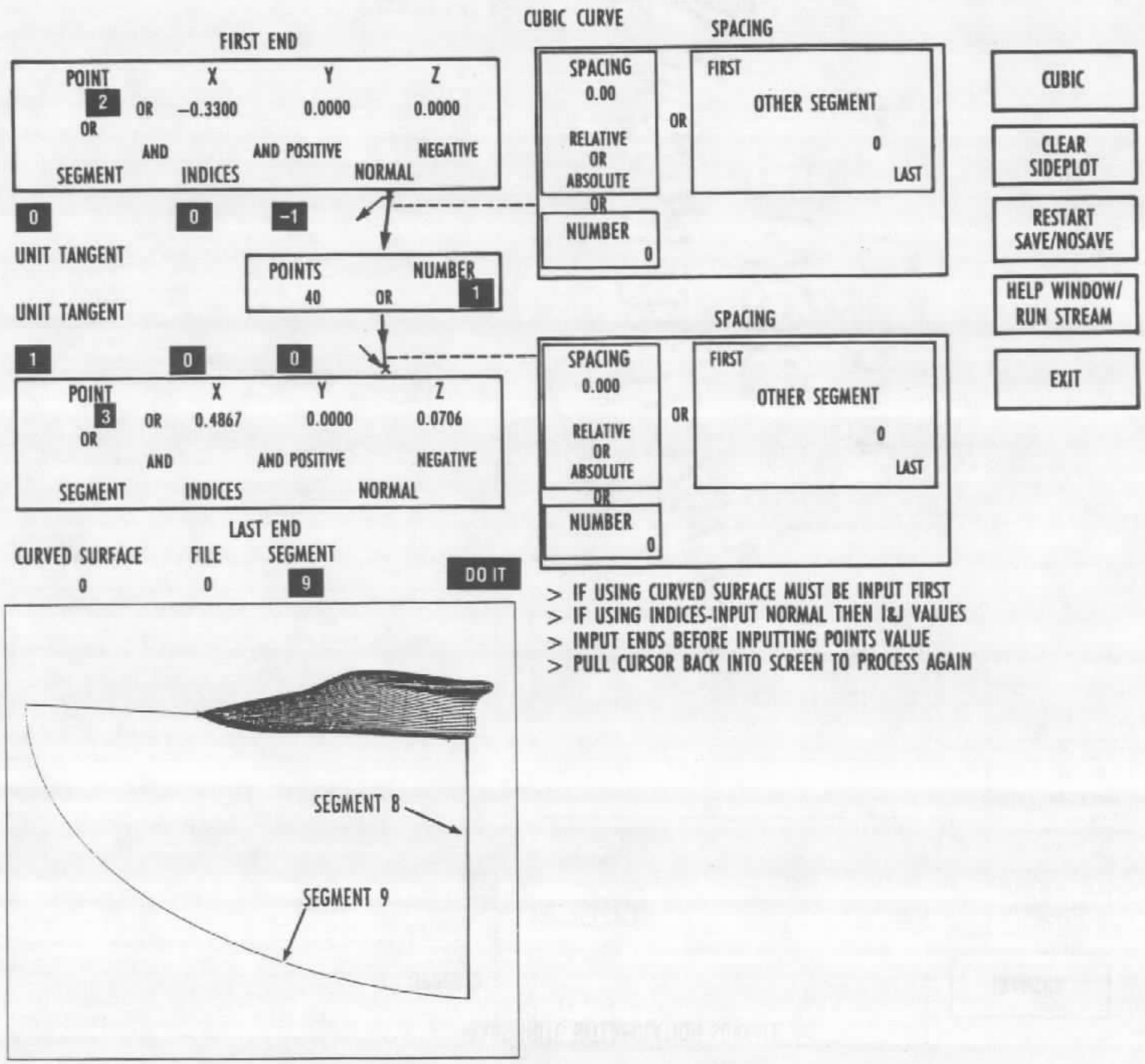


Figure 85. Generation of intersection of outer boundary and symmetry plane.

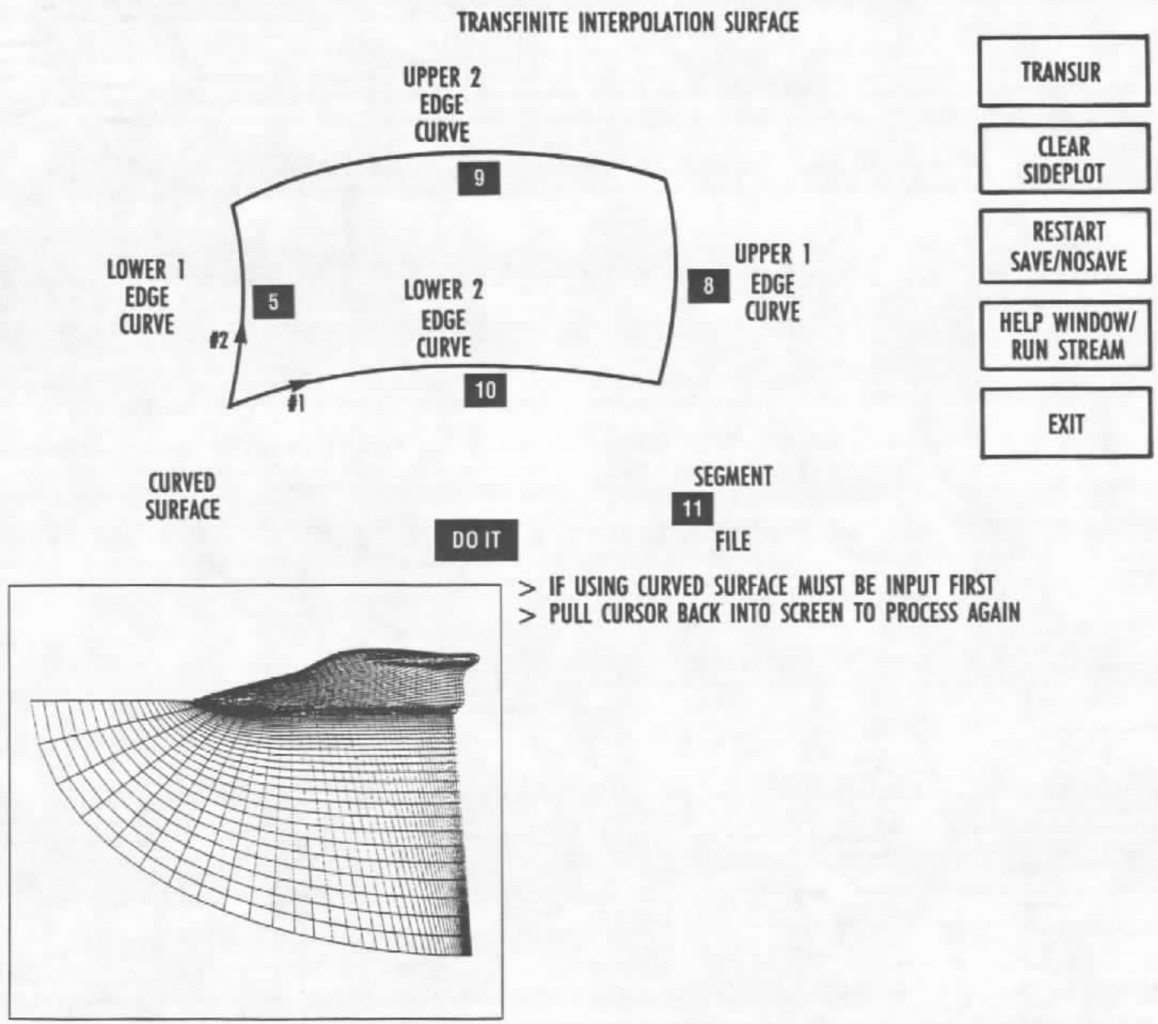


Figure 86. Generation of surface on symmetry plane.

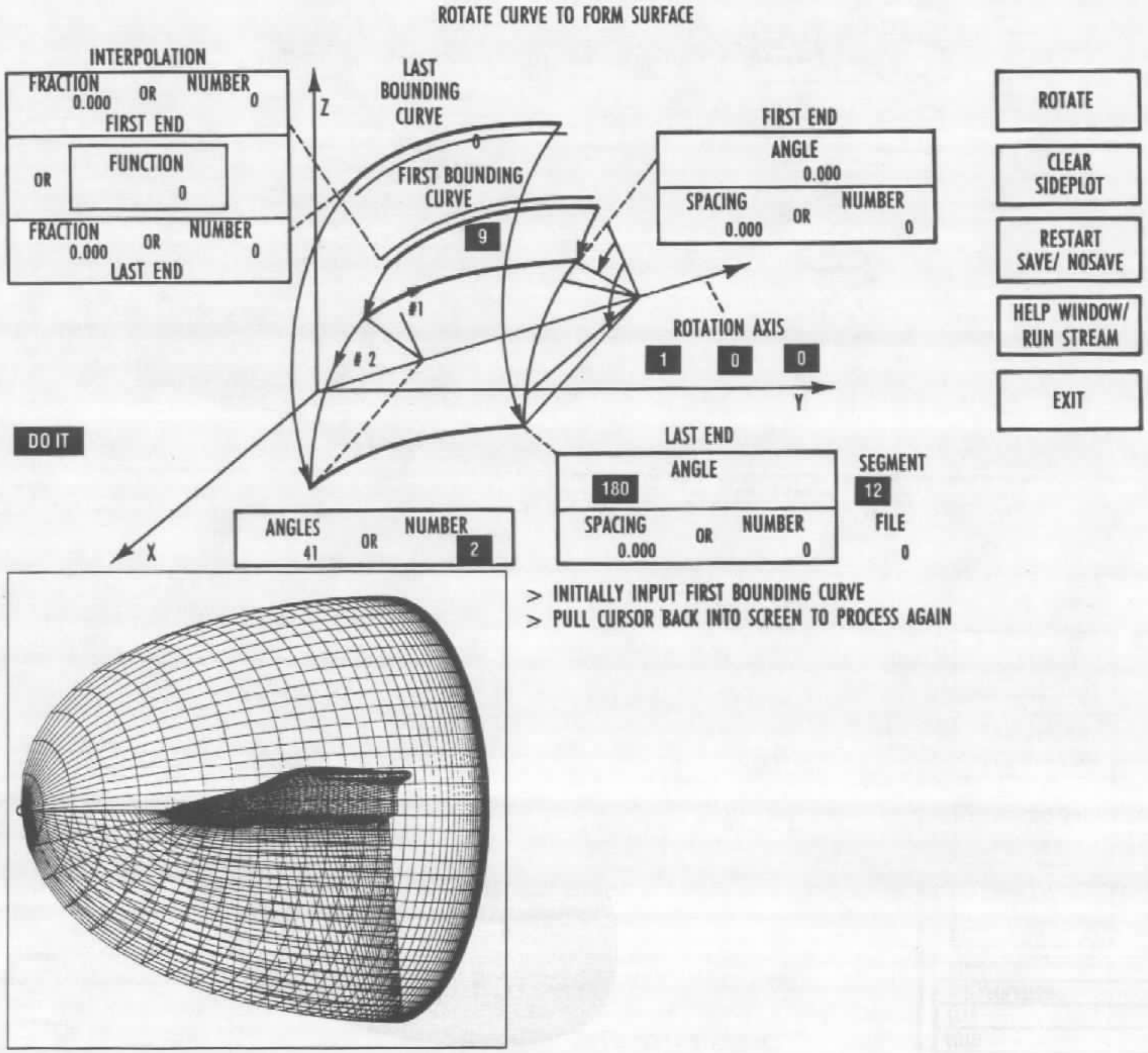


Figure 87. Generation of outer boundary.

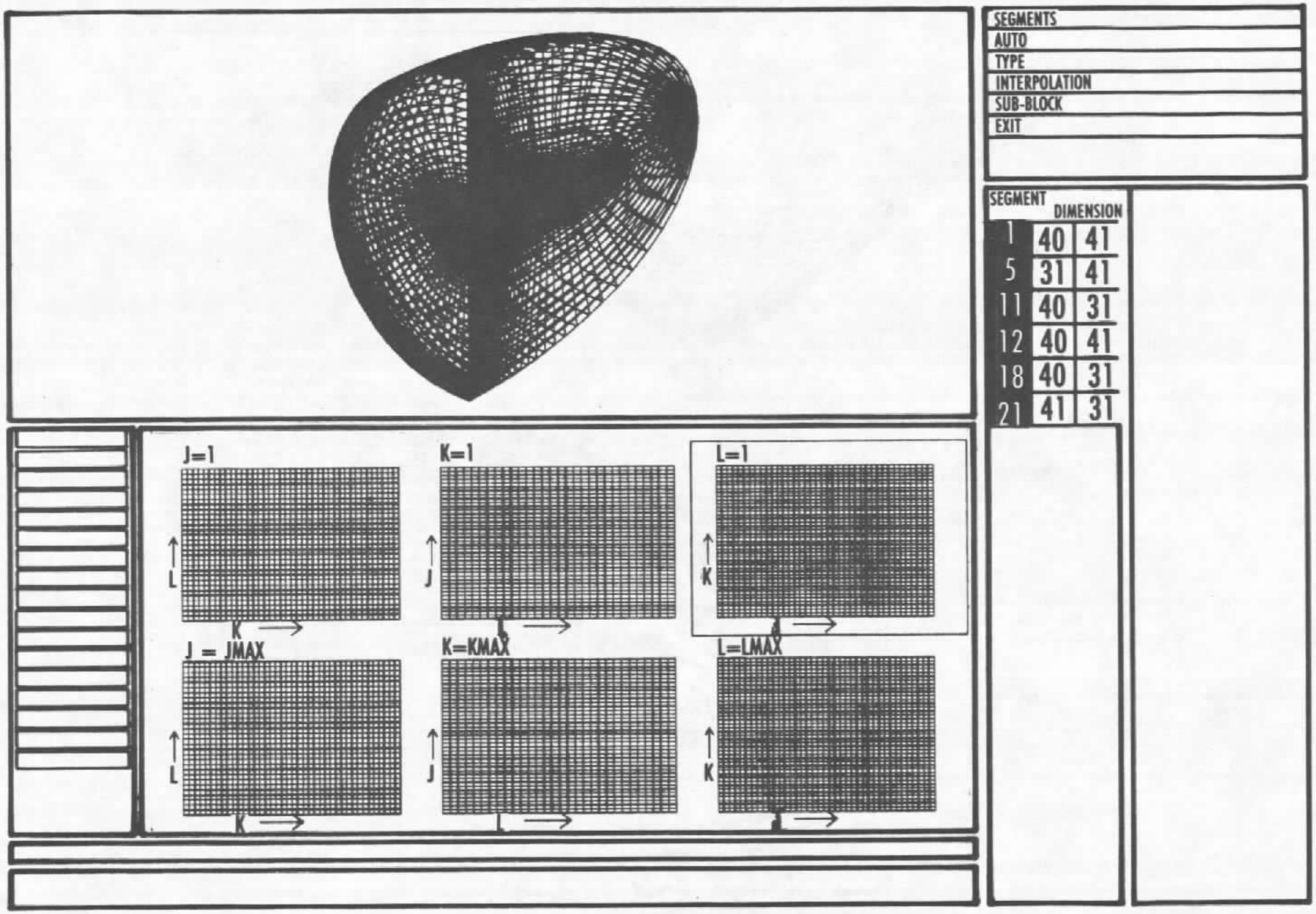


Figure 88. Assignment of physical surfaces to computational boundary.

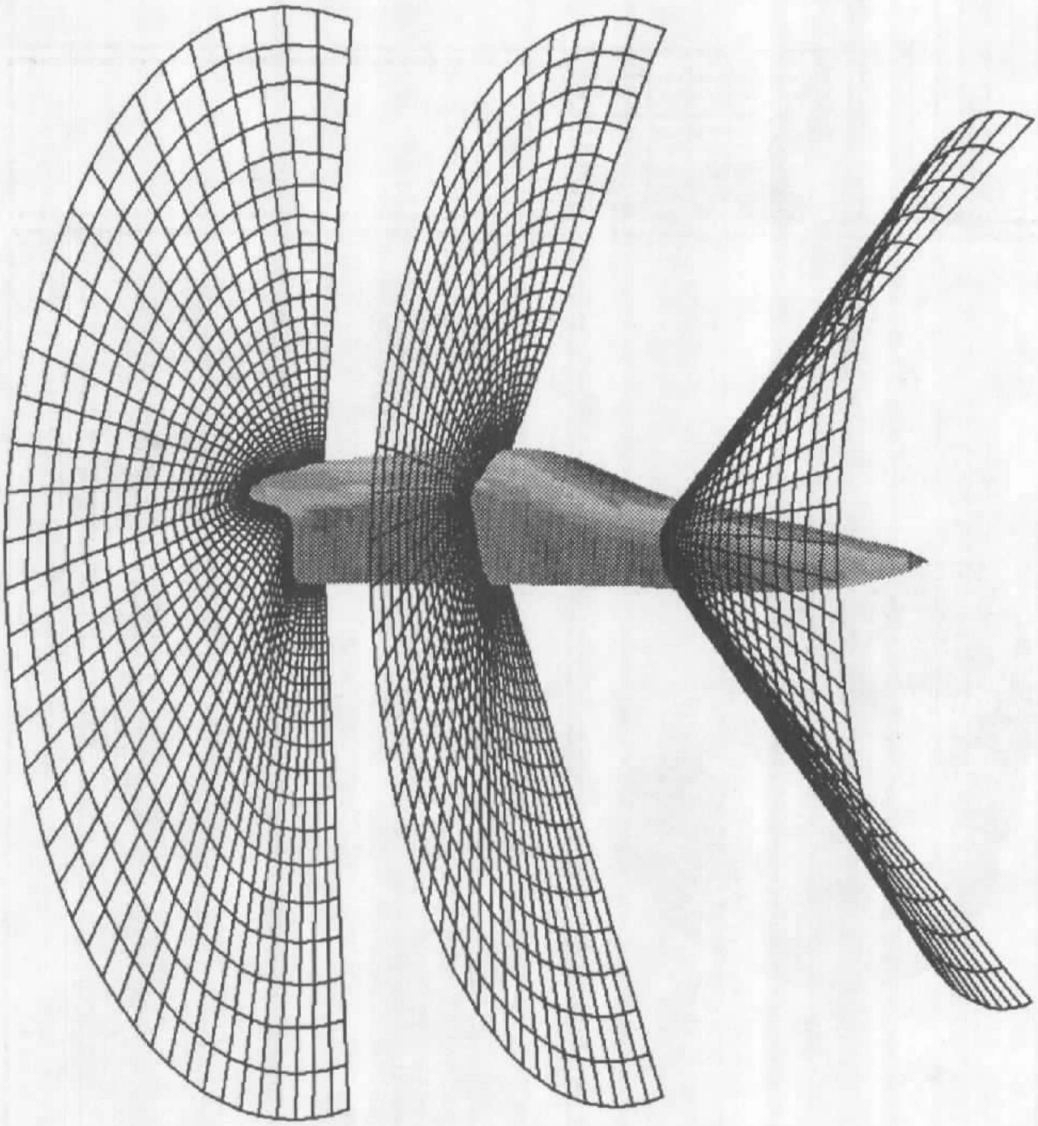


Figure 89. Final three-dimensional grid.