AEROPHYSICS RESEARCH CORPORATION
TECHNICAL NOTE

(GTM) VOLUME I: ENGINEERING DESCRIPTION
AND UTILIZATION MANUAL Final Report
(Aerophysics Research Corp., Houston Tex.)
113 p HC $5.25

By: S. J. Reiners, G. N. Hirsch
G. E. Alford and C. R. Glatt

Prepared for:
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Spacecraft Center
Houston, Texas 77058

December 1974
NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
**Title and Subtitle**
GEOMETRY TECHNOLOGY MODULE (GTM)
VOLUME I - ENGINEERING DESCRIPTION AND UTILIZATION MANUAL

**Author(s)**

**Performing Organization Name and Address**
Aerophysics Research Corporation
Houston, Texas 77058

**Sponsoring Agency Name and Address**
National Aeronautics and Space Administration
Johnson Spacecraft Center
Engineering Analysis Division

**Abstract**
The Geometry Technology Module (GTM) is a system of computerized elements residing in the EDIN (Engineering Design Integration) System library developed for the generation, manipulation, display, computation of mass properties and data base management of panelled geometry. The GTM is composed of computer programs and associated data for performing configuration analysis on geometric shapes. The program can be operated in batch or demand mode and is designed for interactive use.

**Distribution Statement**
Unclassified - Unlimited

**Price**
For sale by the National Technical Information Service, Springfield, Virginia 22151
GTM: GEOMETRY TECHNOLOGY MODULE
VOLUME I - ENGINEERING DESCRIPTION AND UTILIZATION MANUAL

By: S. J. Reiners, G. N. Hirsch
G. E. Alford and C. R. Glatt

Prepared for
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Spacecraft Center
Houston, Texas 77058

December 1974
FORWARD

This final report describing the formulation of the Geometry Technology Module (GTM) is provided in accordance with NASA Contract NAS9-13584. The report is presented in two volumes as follows:


VOLUME II  -  Geometry Technology Module Programmers' Manual.

This work was conducted under the direction of Mr. Robert Abel of the Engineering Analysis Division, National Aeronautics and Space Administration, Johnson Spacecraft Center.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>SURFACE MODEL</td>
<td>7</td>
</tr>
<tr>
<td>Geometric Definition</td>
<td>7</td>
</tr>
<tr>
<td>Geometric Display</td>
<td>7</td>
</tr>
<tr>
<td>Coordinate System</td>
<td>8</td>
</tr>
<tr>
<td>Coordinate Transformations</td>
<td>7</td>
</tr>
<tr>
<td>Mathematical Aids</td>
<td>11</td>
</tr>
<tr>
<td>PROGRAM CHARACTERISTICS</td>
<td>13</td>
</tr>
<tr>
<td>PROGRAM DESCRIPTION</td>
<td>16</td>
</tr>
<tr>
<td>PROGRAM USAGE</td>
<td>20</td>
</tr>
<tr>
<td>Control Cards</td>
<td>20</td>
</tr>
<tr>
<td>Program Input</td>
<td>22</td>
</tr>
<tr>
<td>Master Level Language</td>
<td>22</td>
</tr>
<tr>
<td>Image Input</td>
<td>22</td>
</tr>
<tr>
<td>Input</td>
<td>24</td>
</tr>
<tr>
<td>Cluster Edit</td>
<td>26</td>
</tr>
<tr>
<td>Edit Command Summary</td>
<td>26</td>
</tr>
<tr>
<td>Output Command Summary</td>
<td>27</td>
</tr>
<tr>
<td>Transformation Command Summary</td>
<td>27</td>
</tr>
<tr>
<td>Display Command Summary</td>
<td>28</td>
</tr>
<tr>
<td>Edit Commands</td>
<td>29</td>
</tr>
<tr>
<td>Transformation Commands</td>
<td>33</td>
</tr>
<tr>
<td>Bounding Commands</td>
<td>35</td>
</tr>
<tr>
<td>Register Commands</td>
<td>36</td>
</tr>
<tr>
<td>Miscellaneous Commands</td>
<td>36</td>
</tr>
<tr>
<td>Segment Edit</td>
<td>37</td>
</tr>
<tr>
<td>Point Edit Commands</td>
<td>37</td>
</tr>
<tr>
<td>Segment Level Commands</td>
<td>41</td>
</tr>
<tr>
<td>Program Output</td>
<td>42</td>
</tr>
<tr>
<td>SPECIAL USER INSTRUCTIONS</td>
<td>43</td>
</tr>
<tr>
<td>How to Execute GTM</td>
<td>43</td>
</tr>
<tr>
<td>How to Call a Menu</td>
<td>43</td>
</tr>
<tr>
<td>How to Access Stored Geometry</td>
<td>44</td>
</tr>
<tr>
<td>How to Input Externally Generated Geometry</td>
<td>44</td>
</tr>
<tr>
<td>How to Output Geometry</td>
<td>44</td>
</tr>
<tr>
<td>How to Rotate Geometry</td>
<td>45</td>
</tr>
<tr>
<td>How to Translate Geometry</td>
<td>45</td>
</tr>
<tr>
<td>How to Display Geometry</td>
<td>46</td>
</tr>
<tr>
<td>CONCLUDING REMARKS</td>
<td>46</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>48</td>
</tr>
<tr>
<td>APPENDIX A - GTM COMMAND AND FUNCTION SUMMARY</td>
<td>49</td>
</tr>
<tr>
<td>Master Module Menu Index</td>
<td>49</td>
</tr>
<tr>
<td>Image Input Module Language Index</td>
<td>51</td>
</tr>
<tr>
<td>Cluster Edit Module Menu Index</td>
<td>53</td>
</tr>
<tr>
<td>Segment Edit Module Menu Index</td>
<td>66</td>
</tr>
<tr>
<td>Calculator Module Menu Index</td>
<td>73</td>
</tr>
<tr>
<td>Input Module Menu Index</td>
<td>75</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>GTM DATA BASE</td>
<td>76</td>
</tr>
<tr>
<td>C</td>
<td>SAMPLE PROBLEM</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>Aerodynamic Surfaces</td>
<td>77</td>
</tr>
<tr>
<td>D</td>
<td>SAMPLE PROBLEM</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Gull Wing Geometry Generation</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>Body Enhancement</td>
<td>104</td>
</tr>
</tbody>
</table>
GEOMETRY TECHNOLOGY MODULE (GTM)

VOLUME I - ENGINEERING DESCRIPTION AND UTILIZATION MANUAL


SUMMARY

The Geometry Technology Module (GTM) is a system of computerized elements residing in the EDIN (Engineering Design Integration) System library developed for the generation, manipulation, display, computation of mass properties and data base management of panelled geometry. The GTM is composed of computer programs and associated data for performing configuration analysis on geometric shapes. The program can be operated in batch or demand mode and is designed for interactive use. The significant features of the program are:

1. Data bases containing two and three dimensional shapes including standardized shapes generated by the GTM.

2. An executive computer program containing a user orientated language for controlling the generation, display and calculation of mass properties on selected vehicle components.

3. An auxiliary computer program and data base for the construction and storage of language elements, menus, user instructions and messages.

4. A library of independent geometry generation programs for the creation of specialized geometric panelling.

The techniques used in the program are described including the data base concept. A description of the GTM language and data input requirements is also provided for engineering utilization.
INTRODUCTION

Configuration analysis is considered to be a key element in the design process and one which has not been automated to a large degree. The layout of a candidate vehicle has historically taken place on the drawing board. Geometric input requirements for other technological analysis are derived from the configuration analysis. Figure 1 illustrates the data flow from the configuration analysis technology. The receiving technologies use the geometric data in their analysis and often generate geometric constraints. Many kinds of monitoring diagrams, data displays, drawings, etc. can be and have been generated numerically for rapid display by computer graphics methods. But in the final analysis, the burden of the detailed configuration description has rested with the draftsman under the direction of the design staff. A great deal of experimentation with interactive computer graphics has taken place in an effort to automate the configuration analysis; but their use has had nowhere near the impact that was originally sought. However, recent developments in computer technology have led the way to some reconsideration of the use of interactive graphics in the design process.

1. The advent of the time sharing systems in third generation computers has virtually collapsed the cost of man-machine interface.

2. The development of data base concepts for storage of all design information makes data readily accessible in an online mode.

3. The development of the EDIN system of linking independent programs isolates the graphics function from the analysis modules.

4. The engineering attitude toward direct interface with the computer is changing as a direct result of time-share systems.

A major obstacle in the advancement of interactive graphics technology is the development of a flexible geometry definition module for the layout of aerospace vehicles. It seems evident that a considerable contribution to the automated design process can be made by the GTM.
CONCEPTS

VEHICLE CHARACTERISTICS
MISSION
OPERATIONAL CONSTRAINTS
ECONOMIC GOALS
PERFORMANCE GOALS

CONFIGURATION ANALYSIS

STRUCTURAL GEOMETRY
FUEL LOAD
DEAD WEIGHT LOAD

SURFACE AREAS
VOLUMES
EXTERNAL GEOMETRY
PROTURBANCES
FAIRINGS

ENGINE GEOMETRY
CLEARANCES
INSULATION
APU GEOMETRY

STRUCTURAL ANALYSIS

AERODYNAMIC ANALYSIS

PROPULSION ANALYSIS

FIGURE 1 CONFIGURATION ANALYSIS
Current methods of configuration analysis employ the classical approach to the utilization of automation computing equipment. The GTM method described in this report is contracted with current methods in figure 2. Present day configuration analysis capability, when automated at all, is stratified throughout the engineering community and confined to isolated programs geared to specialized applications. Although the EDIN system offers the designers the ability to collect the analysis capability, the collection does not meet nearly all the needs. More configuration description is usually required. This usually involves additional programming with associated delays in the design analysis.

The GTM method collects the better features of present configuration analysis methods into a single program. It also automates some of the analysis methods not previously available and provides a repository of past configurations to draw upon for future design analysis. In addition, present methods may be employed in conjunction with the GTM analysis through GTM executive commands. The GTM method of configuration analysis is made possible by the data base concepts developed as a direct result of previous EDIN data base development experience.

The basic capabilities exhibited by the GTM are summarized in figure 3. The first Geometry Definition illustrates the user capability to generate new geometry, use existing geometry in standardized formats and use externally generated geometry from another geometry generation program. In conjunction with the user ability to define geometry, GTM provides the user with the capability to manipulate geometry. The manipulation includes translations, rotations, scaling, merging of geometric components, division of geometry and surface fits. GTM also provides for display of geometry. The geometric display can be translated, rotated, overlayed or zoomed for image enhancement. Mass property evaluations may be commanded. The evaluation includes a printout of weights, volume, center of gravity and surface areas of the accessed geometry. The GTM also provides the capability for easily interfacing with the EDIN system.
CURRENT METHODS

INPUT \rightarrow GEOMETRY GENERATION \rightarrow OUTPUT

INPUT \rightarrow GEOMETRY DISPLAY \rightarrow OUTPUT

INPUT \rightarrow GEOMETRY ANALYSIS \rightarrow OUTPUT

GTM METHOD

INPUT \rightarrow GENERATION \rightarrow ENHANCED

DISPLAY \rightarrow EDIN \rightarrow CURVE FITTING

ANALYSIS \rightarrow DATA \rightarrow COMPONENT INTERFERENCE

EXECUTIVE SYSTEM REQUESTS \rightarrow BASE \rightarrow COMPONENT SELECTION

OUTPUT

FIGURE 2  COMPARISON OF CONFIGURATION ANALYSIS METHODS
FIGURE 3  GTM CAPABILITIES.
SURFACE MODEL

The approach taken for exterior geometry consists of generating a data bank of nominal prestored conceptual configuration types. These shapes can include a variety of two and three dimensional shapes recallable from the data base as components or sections of the vehicle, then scaled and smoothed by standardized geometric transformations to provide rapid generation of conceptual vehicle geometry. The resultant shapes can be stored as auxiliary shapes for future recall. This data bank will be capable of ready extension to include additional vehicle types as they become available. The configuration designer can recall a given conceptual design from the data base as his nominal design and display the vehicle on the graphics display. Once the conceptual design is displayed, the vehicle can be scaled, rotated and translated in any manner desired. Components can be selected or deleted from the display to clarify the viewer's interpretation. New components can be generated to replace existing ones. Later versions of the GTM will permit modifications of existing data base geometry. In the interactive mode parameters such as wing sweep, aspect ratio or fuselage fineness ratio will be capable of online modification. By this means, a modified vehicle's external geometry for the conceptual type examined will automatically be generated.

Geometric Definition

The body shape in the GTM is treated as sets of points in three-dimensional space. The basic unit of geometry is the surface point. A grouping of surface points in a plane describes a section. An organization of a number of related sections forms a component. A number of components is used to give a complete description of the vehicle. Each component is an independent unit which can be stored by name and drawn (or displayed) separately or collectively with other components. The geometry can be input in BCD card format generated internal or external to the GTM, retrieved from the data base. Later versions will allow creation at an interactive terminal. The geometry is input to the program in two-dimensional space, section by section. The input technique is similar to a drawing board drafting technique. All data is stored in the data base in section format.

Geometric Display

The display model comes to the stored geometry to quadrilateral elements then scales, translates and rotates the geometry
elements to the desired position and orientation with respect to the viewer. The resulting plot vector is directed to the target display device.

In general, the equations for the geometric rotations described here apply to any orientation angle. The transformations developed apply to rotation of a three-dimensional shape onto a plane. This is precisely the problem for displaying geometry pictorially. The following paragraphs will describe the transformations required for displaying geometry in the GTM.

Coordinate System. - Each point on the surface is described by its coordinates in the body reference coordinate system.

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\] (1)

The body reference coordinate system is assumed to be a conventional right-handed Cartesian system as illustrated below:

![Coordinate System Diagram]

Coordinate Transformations. - To create the perspective drawings or generate geometry, each surface point on the section must be rotated to the desired angle and then transformed into the desired coordinate system. With zero rotation angles the body coordinate system is coincident with the local system in the plane of the paper.
The rotations of the body and its coordinate system to give a desired viewing angle are specified by a yaw-pitch-roll sequence \((\psi, \theta, \phi)\). The rotation is given by the following relationship:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
\phi & 0 & \psi \\
0 & 1 & 0 \\
-\psi & 0 & \phi
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]

When \(X, Y, Z\) represents the coordinates with respect to the system reference axes and \(x, y, z\) represent the coordinates with respect to the local coordinate system, i.e. the plane of the paper or display.
The rotation matrices \( \phi, \theta \) and \( \psi \) are given by:

\[
\begin{bmatrix}
\psi \\
\theta \\
\phi
\end{bmatrix} =
\begin{bmatrix}
\cos \psi & \sin \psi & o \\
-\sin \psi & \cos \psi & o \\
o & o & 1
\end{bmatrix}
\] (3)

\[
\begin{bmatrix}
\theta \\
\phi
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
o & 1 & o \\
\sin \theta & 0 & \cos \theta
\end{bmatrix}
\] (4)

\[
\begin{bmatrix}
\phi
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 \\
o & \cos \phi & \sin \phi \\
o & -\sin \phi & \cos \phi
\end{bmatrix}
\] (5)

or

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
E
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\] (6)

where

\[
\begin{bmatrix}
E
\end{bmatrix} = \begin{bmatrix}
\phi
\end{bmatrix} \begin{bmatrix}
\theta \\
\psi
\end{bmatrix}
\] (7)

Since each point on the surface is given by its coordinates in the \( X,Y,Z \) system, its position in the local coordinate system \( (x,y,z) \) may be found by reversing the above process.

\[
\begin{bmatrix}
x \\
y \\
z
\end{bmatrix} = \begin{bmatrix}
E
\end{bmatrix}^{-1}
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix}
\] (8)

Equation 8 describes the transformation required to rotate vehicle geometry onto the plane of the paper for display purposes. Equation 6 is the transformation required to rotate input geometry to the system reference coordinate system. However, the transformation does not include the translation
of the local coordinate system \( X_0, Y_0, Z_0 \) to the desired position with respect to the system reference axes. The complete transformation is shown below:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix} 
E
\end{bmatrix} \begin{bmatrix} 
X - X_0 \\
Y - Y_0 \\
Z - Z_0
\end{bmatrix}
\]

(9)

Mathematical Aids. - The transformational relationships of equation 8 completely describe the raw transformation required to rotate every point on the vehicle to the desired angle with respect to the reference coordinate system, then onto the plane of the paper. However, the resulting drawing is difficult to interpret because of hidden lines. Further, since only one half, or one quarter (for symmetrical vehicles) of the coordinate points are usually input, some additional calculations are desirable.

Input points are grouped into elements of four in the quadrilateral surface element technique used in this program. The technique provides a convenient means of limiting the drawn lines to those normally seen by the viewer at the defined viewing angles. Each input element is replaced by a plane quadrilateral surface made up of the four lines connecting the points. The quadrilateral characteristics are used to determine the visibility of the four lines. At the users option hidden lines may be drawn or deleted. The quadrilateral characteristics include the area, centroid and the direction cosines of the surface unit normal. The surface unit normals may be transformed through the required rotation angles just as was done for the individual points. The resulting value of the component of the unit normal in the \( X_0 \) direction (out of the plane of the paper) may be found from the following equation:

\[
n_{X_0} = n_x (\cos \theta \cos \phi) + n_y (-\sin \psi \cos \phi + \sin \theta \cos \psi \sin \phi) + n_z (\sin \psi \sin \phi + \sin \theta \cos \psi \cos \phi)
\]

(10)
where \( n_x', n_y', n_z \) are the components of the surface unit normal in the vehicle reference system.

If \( n_x \) is positive then the surface element is facing the viewer. If \( n_x \) is negative the element faces away from the plane of the paper. This result can be used in the program to provide the option of deleting most of those elements on a vehicle that normally could not be seen by a viewer. The resulting picture is thus made more realistic. Confusing elements which are on the back side of a section do not appear. A new criterion will be provided for the deletion of those elements that face the viewer but are blocked by other body sections.

Examination of equation 10 suggests that sections may be further clarified by selectively eliminating elements on the basis of the angle \( \alpha \) that the unit normal makes with the plane of the paper.

\[
\alpha = \cos^{-1}(n_x) 
\]

(11)

If the angle \( \alpha \) is restricted as follows:

\[
0 < \alpha < 5 \text{ (degrees)}
\]

(12)

Then only those panels near the outside envelope of the vehicle would be drawn giving the appearance of a cutaway drawing and permitting the drawings of internal components. This method shall be incorporated in the GTM as a development step in the refinement of the interactive geometry description. The user shall have the ability to describe the angular range of \( \alpha \) for each section of the vehicle.
PROGRAM CHARACTERISTICS

The Geometry Technology Module is applicable to all shell structures internal and external to the vehicle. Geometric analysis performed in the GTM can be augmented by nongeometric data such as mass properties and point mass elements. Major emphasis was the development of a geometric definition module which generates suitable information for use in the EDIN library of technology modules and provides a means of perturbing the geometric definition without a significant impact on manpower or computer resources. The module contains simple weight and balance equations for rapidly accessing the impact of geometric changes on the mass properties of the vehicle. Proper use of the GTM will expose geometric and/or mass properties inconsistencies prior to expending the manpower and computer resources for a full-blown design analysis. The programs are designed for batch, demand or interactive operations and can be easily extended as new capability is developed.

The flexible capability for layout of vehicle external and internal components is available with interfaces with technology program modules for the analysis of aerodynamic, propulsive, weights, structures and performance modules. Geometry Technology Module (GTM) is an independent program set residing in the EDIN library and is executable under control of the DLG program of reference 1. All interprogram communication and geometric data are stored in a large scale data base which can be dynamically constructed under control of the GTM or DLG executive computer programs. The development of the GTM was coordinated with the expansion and extension of the EDIN data base under concurrent development.

The objective in the construction of the EDIN compatible GTM code included:

1. Minimum core size is achieved since a large overlay structure with attendant hard coded interfaces is bypassed. This is particularly significant for a graphics program with current core restrictions.

2. A large data base and a rapid access technique are already available for the storage of geometry data.

3. The geometry generated by the geometry module will be suitable for use in other EDIN programs without transformation.

The characteristics and capabilities of the GTM are illustrated in figure 4. The GTM was developed in such a fashion to enable an engineer to:
GEOMETRY DEFINITION

USER GENERATED GEOMETRY
EXISTING GEOMETRY (STANDARD FORMATS)
EXTERNALLY GENERATED GEOMETRY (FROM ANOTHER PROGRAM)

GEOMETRY MANIPULATION

GEOMETRIC TRANSLATIONS
GEOMETRIC ROTATIONS
GEOMETRIC SCALING
MERGING OF GEOMETRY
DIVISION OF GEOMETRY
SURFACE FITS

GEOMETRY DISPLAY

TRANSLATIONS
ROTATIONS
ZOOMS
OVERLAYS

MASS PROPERTIES

WEIGHTS
VOLUME
CENTER OF GRAVITY
SURFACE AREA

EDIN SYSTEM INTERFACE

GENTRY GEOMETRY OUTPUT
OPERATIONS STACK
GEOMETRIC PARAMETERS

FIGURE 4  GTM CAPABILITIES.
1. Generate interior and exterior configuration geometry by inputting data from a remote terminal and from a data base of stored three dimensional geometrical shapes and standard section data.

2. Perturb configuration geometry by scaling, translation, rotation and mathematical mapping of component geometry.

3. Perturb the configuration by specifying mathematical surface fit functions. Select from a menu of mathematical functions to allow the geometry module to fair the new configuration through the desired shape.

4. Analyze the interior arrangement of geometric components with respect to each other and with respect to the external configuration to avoid volumetric conflict and violation of exterior lines of the vehicle.

5. Select components for the purpose of displaying three dimensional internal and external geometry, cutaways, inboard profiles and composite sections.

6. Select components for the purpose of mass properties evaluation including weight, volume and center of gravity calculations. Specify (non-geometric) point mass properties to be included in this evaluation.

7. Generate the required geometric interface data to be input into the data base which would be consistent with geometry data required for sizing, aerodynamics analysis, packaging, weight and balance, and structural analysis.
The following presents a summary of the physical characteristics of the GTM:

HOST COMPUTER: Univac 1110
FILE NAME(S):
- EX42-00002*GTM2. (SOURCE/RELOCATABLES)
- EX42-00002*GTM. (ABSOLUTE ELEMENT)
- EX42-00002*DATA5. (DATA BASE/LANGUAGE)
- EX42-00002*ODIN-DBINIT2. (MAP ELEMENT)

ABSOLUTE ELEMENT NAME: GTM
LANGUAGE: FORTRAN V
PROGRAM SIZE: 24000 DECIMAL (OVERLAYED)
CARD SOURCE: + 12000
OPERATING MODE: BATCH OR DEMAND
DISPLAY INTERFACE: TEKTRONIX

PROGRAM DESCRIPTION

The GTM uses the Level 2 data management system developed for the Engineering Design Integration (EDIN) System. The system is a set of Fortran callable subroutines which store and retrieve blocks of information on the mass storage media of the Univac 1100 series computers. DMAN maintain a directory of data block names and associated record positions on the mass storage file.

Upon definition of a data block (usually a sequence of X,Y,Z points), a directory entry is dynamically constructed. When DMAN receives a retrieval request, the directory is searched to determine the block location. Data manipulation then takes place on the located data. The GTM is structured to provide the user the capability of manipulating geometry through a hierarchy of subprogram modules.

The executive GTM module is composed of several major executive levels. These levels are called by the GTM executive, named MASTER. The major executive levels are the input module, cluster edit module and segment edit module. Figure 5 illustrates the GTM program organization.
FIGURE 5  GTM PROGRAM ORGANIZATION.
The MASTER module (GTM Executive) is the control point in the GTM from which all sublevel executives are accessed. It contains its own language set which allows the user to perform database management functions, access sublevel executives and general program control. Three primary sublevel languages are available, input, segment edit and cluster.

The INPUT sublevel executive is provided for reading data which is stored in specific geometry formats. Two are available, the Gentry format of reference 1 and the GTM format. GTM format allows free-field data to be entered. The data may be any type of information. This data is read in and stored in the database geometry tree structure. The INPUT module contains its own language set and associated menus, which can be displayed upon command.

The CLUSTER EDIT Module contains a language subset and instructions necessary for creating and maintaining the geometric data tree structure. Functions are also provided for translation, rotation and scaling of tree stored data and output of the data in forms for interfacing with other EDIN technology modules. In addition, it contains the necessary logic to display geometry for image viewing. The display functions have a number of features which allow the user to zoom in on a specific region, overlay geometry, scale geometry and filter geometry for resolution. Mass properties evaluations are also commanded from the CLUSTER EDIT Module.

The SEGMENT EDIT Module provides the capability to compose geometric shapes, manipulate geometry at the segment level and display of geometric segments. Specific operations include translation, rotations, scaling, point redistributions, segment cutting, point edit commands and display. The module contains its own language subset addressable by the user.

The GTM provides the capability of maintaining and updating geometry information in a name oriented data base. The geometry can be a section, component or a cluster, as shown in figure 6.

A section is defined as a sequence of arbitrary X,Y,Z points defining a line in three dimensional space. A component is a collection of sections approximating surface points in three dimensional space. A cluster is defined as a collection of components which form a complete or partial surface configuration. The data can be tree structured at the cluster level so analysis can be performed on groups or collections of data with relatively simple data structure definitions. Once the data is assessed by the GTM, a variety of manipulation techniques are available at all data definition levels through the GTM language.
FIGURE 6  GTM GEOMETRY STRUCTURES.
When geometry is inserted into the data base in a tree structure, the tree can be defined at four levels as illustrated by the following:

- CLUSTER NAME
- COMPONENT NAME
- SECTION NAME
- DATA POINT

Individual geometry data sets are generally stored only once though each set may belong to more than one cluster (tree).

The GTM has advanced to the state where geometry can be stored by name at several hierarchical levels in a tree structure. Editing of the data can be performed at all levels of data definition. The program contains the basic utilities which permit the development of user orientated manipulation of geometry and critical manipulation functions such as scaling, rotation and translations.

PROGRAM USAGE

The computer program usage requirements described in this section are oriented toward the Univac Exec 8 1110 version and specifically towards the Johnson Spacecraft Center's installation. The actual program input (language commands) described are applicable wherever the program is installed but the control cards of the program will differ from computer to computer.

Control Cards

The control cards for execution of the GTM are illustrated by figure 7. After input of the run card, an assign of temporary file one (1) is required for data base storage. The data base and associated GTM language set presently resides on file EX42-00002*DATA5. Instructions for creating a new data base and the associated language structure are contained in reference 4. A copy of the DATA5 file to temporary file one (1) is required to protect the integrity of the data base. All I/O is stored and retrieved from file 1 during execution. If the user wishes to retain any data base entries during execution, he may permanently save these entries by a copy of file 1 to DATA5 prior to termination. Following the execute command, the user is free to enter GTM orientated commands. A brief discussion of these commands follows. Further information on the command structure and associated language set can be found in Appendix A.
FIGURE 7  TYPICAL RUN STREAM.
Program Input

The GTM operates upon a database of stored information which is manipulated by a language set which is the input to the program. The information which follows describes in brief the language structure of the GTM.

The GTM consists of a master level executive and other lower level executives. The executives are responsible for executing a specific task upon request by the user. A language consisting of manipulation commands for controlling the GTM at each executive level is available.

Master Level Language

The following commands are the statements available in the master executive at the present time:

*IMAGE INPUT
*INPUT
*CLUSTER EDIT
*SEGMENT EDIT
SAVE DATA BASE (_)
OPS STACK (_:_)
MENU
EXIT
STOP

*Sublevel executives

Image Input

This command will cause a transfer to the Image Input Executive. The Image Executive is provided as a means of reading data which is stored in the arbitrary body coordinate (IMAGE Program) format of reference 2. This data is read in and stored in the database geometry tree structure.
General Commands:

MENU - Provides a list of available commands at the current language level.
EXIT - Returns control to the master level language.

Data Source Commands:

DATA BASE (___:___:___) - The data resides in the data base in the card image form.
BCD FILE n - The data is a file n and is formatted data.
BINARY FILE n - The data is a file n and is unformatted or binary data.

Tree Structuring Commands: The IMAGE formatted data contains status codes of 0,1,2,3. All points are considered status 0 except as follows:

Status 1 Beginning of a new section.
Status 2 Beginning of a component or subcomponent (synonymous in the GTM)
Status 3 End of a cluster of components.

The status flags are used when entering the data into the GTM data base to position the data within the geometry structure. For each Status 3, or the beginning or a file of input, the following command must be input:

[CLUSTER] = ___:___:___

This command will cause all subsequent data with status less than 3 be entered into the tree under the name ___:___:___. For each Status 2 encountered, the following command must be input:

[C COMPONENT] = ___:___:___

This command will cause all subsequent data with Status 2 to be stored in the data base geometry tree structure under this name.
NOTE: For Status 1, the section names are set by default only. The default names are:

SECTION 1
SECTION 2

for each Status 1 encountered in the component.

Input

This command will cause a transfer to the free-field data and can be used to enter data blocks and any type of information. The input block has the following requirements:

HEADER This statement gives the type of storage and the name under which the data is to be entered into the data base.

END This statement signifies the end of the input block.

Types of Header Statements:

BCD INPUT (: :)
This header is used to store the card image data in the data base.

NUMERIC INPUT (: :)
This header is used to enter numeric data into the data base. The data is read in a free-field format. The values on the card must be separated by delimiters, which can be either a space or a comma. Commas back to back specify null fields between them.
OPS STACK

This statement is used to input an OPS Stack to the data base. An OPS Stack is an instruction string of commands which can be executed by using the OPS STACK command.

Input items can be read from other files using the READ FILE n command. This will cause the read to transfer to the specified file and continue with that file until the file is exhausted.
Cluster Edit

The Cluster Edit language subset contains instructions necessary for creating and maintaining the geometric data tree structure. Functions are also provided for translation, rotation, display and scaling of tree stored data and output of the data in forms for interfacing with other programs.

Edit Command Summary.

<table>
<thead>
<tr>
<th>Command</th>
<th>Subcommand</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILD</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>BUILD</td>
<td>[SEGMENT]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>ACCESS</td>
<td>[CLUSTER COMPONENT] [SECTION] [SEGMENT]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>LOCATE</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>INSERT</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>DELETE</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>REPLACE</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>COPY</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>COPY</td>
<td>[SEGMENT]</td>
<td><em><strong>:</strong></em></td>
</tr>
<tr>
<td>ADD</td>
<td>[CLUSTER COMPONENT] [SECTION]</td>
<td><em><strong>:</strong></em></td>
</tr>
</tbody>
</table>
Output Command Summary. -

[LIST] [CLUSTER COMPONENT SECTION] (_:_:_)

TREE LIST (_:_:_)
LIST AVAILABLE CLUSTERS
COPY BINARY (_:_:_)
COPY BCD (_:_:_)

Transformation Command Summary. -

Rotation

[PSI YAW] = ___
[THETA PITCH] = ___
[PHI ROLL] = ___
[RCEN ROTATION CENTER] = _'_'
[RUEC ROTATION VECTOR] = _'_'
[ROT ROTATE]

Scaling

MAG = ___
XMAG = ___
YMAG = ___
ZMAG = ___
[MAGC MAGNIFICATION CENTER] = _'_'
[SC SCALE]

ORIGINAL PAGE 3
OF POOR QUALITY
Display Command Summary:

- DISPLAY ____ : ____ : ____
- DISPLAY + ____ : ____ : ____
- DISPLAY - ____ : ____ : ____
- REFRESH
- AFILT = ____
- RFILT = ____
- PSI = ____
- THETA = ____
- PHI = ____
- SYM
- NOSYM
- SCALE = ____
- ZOOM = ____ , ____

Translation

- XMOVE = ____
- YMOVE = ____
- ZMOVE = ____
- MOVE

Bounding Commands

- START ____ : ____
- STOP ____ : ____

Register Commands

- ZERO [BUILD ]
  [ACCESS ]
  [LOCATE ]
- ACCESS [BUILD ]
  [LOCATE ]
- BUILD [ACCESS ]
  [LOCATE ]
- LOCATE [ACCESS ]
  [BUILD ]
Edit Commands. Addressing a tree structure requires the maintenance of a list of data pointers, one for each level of the tree. These lists are called registers. Three registers are maintained in the GTM, a build, an access and a locate register. The build register is constructed by the GTM when the geometry is initially stored. The build register can be thought of as the output register. For instance, data is copied from the access register to the build register. The access register is used when a geometric manipulation is performed. The access register can be thought of as the input register, although many commands affect the data in this register. The locate register is a temporary register used when data is being transferred or modified. The locate register is not saved from command to command. It is zeroed after each use. It is used as a working register by other executive functions so it must be reestablished implicitly (by the program) or explicitly (by the user) prior to its use. The edit commands are used to initially establish the register as well as to maintain the actual data referenced by these registers.

The register contents are used to control the limits of action of such statements as copy, move and rotate. There is an entry in the register for each tree level. The specified action such as exemplified above begins sequentially at the first significant (non-zero) entry, and proceeds for all data below that level. Thus, if only the cluster entry is non-zero, the specified action will take place on the entire cluster. If a component is specified, the action will apply to that component only. If a section is specified, the action will apply only to that section.

The edit commands must proceed in a hierarchical manner. A cluster must be referenced before a component, and a component must be referenced before a section.
This command provides for the creation of a new entry at the level where it is applied. If an entry by the same name already exists at this level, the older entry will be destroyed and new entry will replace it. This command can be used only for the creations of new entries (see REPLACE). The BUILD command is used to maintain the BUILD register.

**BUILD SEGMENT**

This command, although similar to the preceding commands, is not a tree structure command because it is manipulating geometry at the lowest level data structures (the segment). The command will cause a new title to be defined in the data base in preparation for the receipt of a data block representing a sequence of X,Y,Z coordinates. SEGMENT EDIT Commands which follow will perform the actual data structuring.

**ACCESS**

This command is used as a prelude to the manipulation of geometry within a tree structure. It establishes a sequence of pointers called the access register which identifies the geometry to be manipulated.

The ACCESS Command actually provides for the redefinition of the contents of the ACCESS Register. This register contains information which determines data to be copied. For example, the register determines the insertion position and the replacement position for INSERT and REPLACE Commands. The contents of this register also controls the data to which transformations are applied.

**LOCATE**

This command is used as a prelude to the use of data associated with the locate name. For example, to copy or insert data from one tree structure, one would use the LOCATE Command prior to the COPY Command.

This command actually maintains the Locate Register which is essentially a temporary set of pointers for the purpose of buffering data into a tree structure controlled by the Access Register. It is used primarily by the INSERT and REPLACE Commands. If the item to be transferred using
an INSERT or REPLACE Command is itself a resident of a tree, the LOCATE Commands must be used to define the data prior to the command execution.

\textbf{INSERT} \quad \begin{array}{c}
\text{CLUSTER} \\
\text{COMPONENT} \\
\text{SECTION}
\end{array} \quad (_{_::_})

This command will cause a new cluster component, or section to be inserted into a geometric data tree structure. The position of the insertion is defined by the Access Register and will be the position \textit{in front} of the position specified by the access.

\textbf{NOTE:} If the data to be inserted is part of the tree, the named \( (_{_::_}) \) title field must be replaced by a proper series of LOCATE Commands.

\textbf{DELETE} \quad \begin{array}{c}
\text{CLUSTER} \\
\text{COMPONENT} \\
\text{SECTION}
\end{array} \quad (_{_::_})

This command will cause the specified item to be deleted from the tree. The Controlling Register is the Access Register. If the name \( (_{_::_}) \) title field is omitted, the item deleted will be the item at the level specified and defined in the Access Register.

\textbf{REPLACE} \quad \begin{array}{c}
\text{CLUSTER} \\
\text{COMPONENT} \\
\text{SECTION}
\end{array} \quad (_{_::_})

This command will cause one item to be replaced by another. If the new item is resident in another tree structure, the named \( (_{_::_}) \) field must be preceded by an appropriate series of LOCATE commands.

\textbf{COPY} \quad \begin{array}{c}
\text{CLUSTER} \\
\text{COMPONENT} \\
\text{SECTION}
\end{array} \quad (_{_::_})

This command will cause the specified data to be physically copied from its current data source into the tree structure specified by the BUILD Command. If a BUILD Command was executed prior to the COPY Command, the highest level copied will have its title changed to the title given on the BUILD Command. All other titles will remain unchanged. If BUILD Command was not given, all titles remain unchanged.

\textbf{COPY SEGMENT} \quad (_{_::_})
This command can be used to copy segments into a tree structure as sections. The title of the section must be specified by a BUILD SECTION Command. Several segments can be copied into a single section by executing more than one COPY SEGMENT Command before executing a BUILD SECTION Command.

Segments may be copied into segments by executing a BUILD SEGMENT Command prior to the COPY SEGMENT Command.

The SEGMENT EDIT sub-language executes transformations which can not be performed on tree structured data. Data stored in the tree structure must be copied to segments before the Segment Edit functions can be performed. This is done by first executing a BUILD SEGMENT Command and then executing a COPY SECTION Command, and repeating for each section for which Segment Edit functions are desired.

```
ADD
[ CLUSTER COMPONENT ]
[ SECTION SEGMENT ]
```

This command is equivalent to a COPY Command except the data is not physically copied. Only pointers are transferred so that the proper tree linkages are established. If the (___:___:) title field is omitted, the current Access Register is used to control the command. Titles of components and sections can not be changed by using the BUILD Command prior to an ADD Command; this would result in an error.

Output Commands

```
LIST
[ CLUSTER COMPONENT ]
[ SECTION SEGMENT ]
```

This command will cause the contents of the specified tree level, or item, to be listed. Thus, LIST VEHICLE provides a list of all components in a vehicle. LIST COMPONENT provides a list of all sections in the component. LIST SECTION or LIST SEGMENT provides a listing of all points in the section or segment.

```
TREE LIST
( ___:___:___ )
```

32
This command will cause the entire tree structure of the specified vehicle to be listed.

LIST AVAILABLE CLUSTERS

This command will cause a listing of the vehicles available in the data base to be listed.

COPY BINARY (___:___:___)
COPY BCD (___:___:___)

These commands will cause the specified vehicle to be output in the IMAGE format of reference 1. The first command causes a binary file to be written. The second command causes a BCD file to be written.

If less than a full cluster is desired, the (___:___:___) field must be omitted and the output item established by the appropriate ACCESS Commands. START and STOP apply to this command.

Transformation Commands. - These commands use the right hand coordinate system with x position forward.

Rotation Parameters:

\[
\begin{bmatrix}
\text{PSI} \\
\text{YAW}
\end{bmatrix} = ___
\]

This is the yaw angle in degrees desired for this rotation.

\[
\begin{bmatrix}
\text{THETA} \\
\text{PITCH}
\end{bmatrix} = ___
\]

This is the pitch angle in degrees desired for this rotation.

\[
\begin{bmatrix}
\text{PHI} \\
\text{ROLL}
\end{bmatrix} = ___
\]

This is the roll angle in degrees desired for this rotation.

\[
\begin{bmatrix}
\text{RCEN} \\
\text{ROTATION CENTER}
\end{bmatrix} = ___,'___'
\]
These are the X,Y,Z coordinates of the center of rotation.

\[
\begin{bmatrix}
\text{RVEC} \\
\text{ROTATION VECTOR}
\end{bmatrix} = \_\_\_\_\_\_
\]

This is an alternate way of inputting the rotation angles.
In this case, it is a rotation vector with I,J,K input.
This does not need to be a unit vector.

\[
\begin{bmatrix}
\text{ROT} \\
\text{ROTATION COMMAND}
\end{bmatrix}
\]

This command will cause the specified item to be rotated.
The data to be transformed must have been established by
an appropriate set of ACCESS Commands. The desired rota-
tion parameters must have been established before this
command is issued. The default parameters for all para-
eters are zero. The values established are saved and
need not be changed for subsequent and identical rotations.

Scaling Parameters: - The parameter commands are:

\[ \text{MAG} = \_\_\_\_\_\_\_ \]

This command causes the X,Y,Z scale factors to be set to
the same value.

\[ \text{XMAG} = \_\_\_\_\_\_\_ \]

This command sets the X scale factor.

\[ \text{YMAG} = \_\_\_\_\_\_\_ \]

This command sets the Y scale factor.

\[ \text{ZMAG} = \_\_\_\_\_\_\_ \]

This command sets the Z scale factor.

\[
\begin{bmatrix}
\text{MAGC} \\
\text{MAGNIFICATION CENTERS}
\end{bmatrix} = \_\_\_\_\_\_\_\_\_
\]

The scaling equations used are:

\[ \text{XOUT} = (\text{XIN} - \text{XC}) \times \text{XMAG} + \text{XC} \]
\[ \text{YOUT} = (\text{YIN} - \text{YC}) \times \text{YMAG} + \text{YC} \]
\[ \text{ZOUT} = (\text{ZIN} - \text{YC}) \times \text{ZMAG} + \text{ZC} \]
This allows the scaling or magnification to take place about a specific point. The center of magnification (XC, YC, ZC) is input by this command.

\[
\text{SC} \\
\text{SCALING COMMAND}
\]

This command causes the specified scaling to be executed. The item to be scaled must have been established by a preceding set of appropriate ACCESS Commands.

**NOTE:** The default values for scale factors are 1.0, 1.0, 1.0 and 0.0, 0.0, 0.0 for the magnification center.

**Translation Parameters.**

XMOVE = ____

The X translation distance.

YMOVE = ____

The Y translation distance.

ZMOVE = ____

The Z translation distance.

\[
\text{MOVE} \\
\text{TRANSLATION COMMAND}
\]

This command will cause the translation to occur. The item to be translated must have been established by a preceding set of appropriate ACCESS Commands.

**NOTE:** The default translation values are 0,0,0.

**Bounding Commands.**

START ____:____:____

STOP ____:____:____

These commands allow a Start and Stop position to be specified for a given operation. If the item being operated on is a cluster, these can be component names. If the item being operated on is a component, these can be section names.

The operation specified included the start position, the stop position and all items between.
The Start and Stop Registers are nulled after each use.

These commands can be used prior to the following commands to identify geometry to be manipulated.

COPY
LIST
COPY BINARY
COPY BCD
ALL TRANSFORMATIONS

Register Commands.

ZERO

\[
\begin{bmatrix}
\text{BUILD} \\
\text{ACCESS} \\
\text{LOCATE}
\end{bmatrix}
\]

This command causes the specified register to be set to Zero.

\[
\begin{bmatrix}
\text{BUILD} \\
\text{ACCESS} \\
\text{LOCATE}
\end{bmatrix}
= \begin{bmatrix}
\text{BUILD} \\
\text{ACCESS} \\
\text{LOCATE}
\end{bmatrix}
\]

These commands allow the contents of one register to be transferred to the other specified register.

Miscellaneous Commands.

MENU - List a menu of the available commands.

OMIT - Exit the section.

EXIT - Return to the next highest language.
Segment Edit

A segment is a sequence of X,Y,Z coordinates in three dimensional space. They are distinguished from sections in that they are not part of a data tree structure as in the case of the section. Each segment is resident in the data base under its own unique title. Therefore, any transformation can be executed on the data including transformations which increase or decrease the number of data points. The number of data points must remain the same for point level data in the geometric data tree structure.

Point Edit Commands. - The Point Edit Commands are those which apply to a single point of data. Since the definition of the internal data is the ordered set of coordinate points, each data point has 3 values and has an implied point number corresponding to the order in which it was placed in the data base. This point number is used to establish the action position for future point edit commands. The Point Edit Commands are:

\[
\begin{align*}
\text{P} & \text{POINT} = n \\
\text{F} & \text{FIND} (X,Y,Z) \\
\text{R} & \text{REPLACE} (X,Y,Z) \\
\text{I} & \text{INSERT} (X,Y,Z) \\
\text{D} & \text{DELETE} (X,Y,Z) \\
\text{A} & \text{DEFINE} (X,Y,Z) \\
\end{align*}
\]

\[
\begin{align*}
\text{P} & \text{POINT} = n: \text{This command defines a point number where some subsequent action may be performed on a segment. It is referred to as the action position within the segment.} \\
\text{F} & \text{FIND} (X,Y,Z): \text{This command will locate the action position or point number of the point in the segment nearest the specified}
\end{align*}
\]
Each action command has an optional title field associated with it. Each action requires two titles; the title of the segment to process and the title under which to store the processed segment.

The title of the segment to process is either the input title or the title of the last segment output, or the title of the last segment accessed, in that order.

The only way to specify an output title different from the input title is to use the BUILD Command. This command will set the output title. Immediately after execution of an action, the input title is reset to the output title, so that this title then becomes the default title.

Limit Definitions: The commands are:

\[
\begin{align*}
\text{START} & \quad X, Y, Z \\
\text{STOP} & \quad X, Y, Z \\
\text{NSTART} & = \quad n \\
\text{NSTOP} & = \quad n
\end{align*}
\]

These commands determine the limits between which a given transformation is to take place. START and STOP use FIND to determine NSTART and NSTOP, the first and last point numbers.

Translation: The commands are:

\[
\begin{align*}
\text{XMOVE} & = \quad \_\_\_ \\
\text{YMOVE} & = \quad \_\_\_ \\
\text{ZMOVE} & = \quad \_\_\_ \\
\text{MOVE} & = (\_\_:\_\_:\_\_)
\end{align*}
\]

XMOVE, YMOVE and ZMOVE set up the translation distances. The default values are zero. The command MOVE causes the actual translation to take place.

Scaling: The commands are:

\[
\begin{align*}
\text{XMAG} & = \quad \_\_\_ 
\end{align*}
\]
XMAG, YMAG and ZMAG are the magnification factors applied during the scaling operations. MAG sets all of the magnification factors to the same value. The default magnification values are 1.0. The transformation equations are:

\[
X_T = (X_I - X_C) \times X_{MAG} + X_C
\]

\[
Y_T = (Y_I - Y_C) \times Y_{MAG} + Y_C
\]

\[
Z_T = (Z_I - Z_C) \times Z_{MAG} + Z_C
\]

The Command MAGNIFICATION CENTER = X_C,Y_C,Z_C establishes the values as X_C,Y_C,Z_C. The default values are zero. The Command SCALE causes transformation to be executed.

Rotation: The ROTATIONS Commands are:

\[
PSI = \text{___} \quad \text{(Degrees)}
\]

\[
YAW = \text{___} \quad \text{(Degrees)}
\]

\[
THETA = \text{___} \quad \text{(Degrees)}
\]

\[
PITCH = \text{___} \quad \text{(Degrees)}
\]

\[
PHI = \text{___} \quad \text{(Degrees)}
\]

\[
ROLL = \text{___} \quad \text{(Degrees)}
\]

\[
R_CEN = \text{___} \quad \text{(Degrees)}
\]

\[
ROT = \text{___} \quad \text{(Degrees)}
\]

YAW, PITCH and ROLL establish the rotation angles of the transformation. The default values are zero. ROTATION CENTER
establishes the center of rotation of the transformation. The Command ROTATE causes the transformation to be executed.

Cutting a Segment with a Plane:

**Plane Definition:** The Plane may be defined and used in the rotation command. In this case RCEN is a point on the Plane, and PSI, THETA and PHI are angles describing the direction of the normal.

\[
\text{PLANE} = X_C, Y_C, Z_C, \text{PSI}, \text{THETA}, \text{PHI}: \text{ This is an one-line command setting all the values described above.}
\]

\[
\text{Xcut} = \_\_\_: \text{ This assumes a YZ Plane passing through the specified X with a direction of positive X.}
\]

\[
\text{Ycut} = \_\_\_: \text{ This assumes a XZ Plane passing through the given Y value in the direction of positive Y.}
\]

\[
\text{Zcut} = \_\_\_: \text{ This assumes a XY Plane passing through the given Z with a direction of positive Z.}
\]

\[
\text{Pcut (\_\_\_: \_\_\_): This command will cause all of the points in the direction of the positive normal, plus all plane intersections to be output.}
\]

\[
\text{Mcut (\_\_\_: \_\_\_): This command will cause all of the points in the direction of the negative normal, plus the plane intersections to be output.}
\]

**Point Redistribution:** The commands are:

\[
\text{NSEG} = \_\_\_\_\_\_\_
\]

\[
\text{EQLEN (\_\_\_: \_\_\_)}
\]

These commands will cause the segment to be redistributed such that the arc length of the line described by this space is divided into NSEG equal positions. This means that NSEG + 1 points are output to describe this point redefinition.

**Data Acquisition:** Four CLUSTER EDIT Commands are included to allow the user access to tree stored section data. These commands are:

\[
\text{\*ACCESS CLUSTER \_\_\_: \_\_\_}
\]

\[
\text{\*ACCESS COMPONENT \_\_\_: \_\_\_}
\]
point such that \((X_S-X_F)^2 + (Y_S-Y_F)^2 + (Z_S-Z_F)^2\) is a minimum.

In some cases there will be multiple points of the same value. For example, a cross section closing on itself will have identical first and last points. This situation is handled by the use of additional calls to FIND. If FIND already has been called, then the search for the point begins at the next point after the one found by the previous command.

\[
\text{REPLACE} \quad (X,Y,Z): \quad \text{This command will cause the point specified by action position to be replaced by the given } X, Y, Z \text{ values. If omitted, the point specified by the DEFINE Command will be used.}
\]

\[
\text{INSERT} \quad (X,Y,Z): \quad \text{This command will cause the given } X, Y, Z \text{ values to be inserted in front of the action position. If } X, Y, Z \text{ is omitted, the point specified by the DEFINE Command will be used.}
\]

\[
\text{DELETE} \quad (X,Y,Z): \quad \text{This command will cause the } X, Y, Z \text{ input point to be deleted, and if omitted, the point specified by the action position will be deleted.}
\]

\text{NOTE: If the } X, Y, Z \text{ is input, a procedure similar to FIND is used to determine the point to be deleted.}

\[
\text{DEFINE} \quad (X,Y,Z): \quad \text{This command can be used in place of the } X, Y, Z \text{ inputs in all of the point commands except POINT and DELETE.}
\]

\[
\text{ADD} \quad (X,Y,Z): \quad \text{This command will cause the input } X, Y, Z \text{ values to be added to the end of a segment. If } X, Y, Z \text{ is omitted, the point defined by the DEFINE Command will be used.}
\]

\text{NOTE: The values determined by POINT and FIND are zeroed after each use. They are not maintained.}

\text{Segment Level Commands.} - \text{Segment level commands are those which apply or transform to an entire segment. The commands fall into two groups - informational and action.}
ACCESS SECTION __:__:

COPY SECTION (__:__):

These commands will allow the user to construct segments from existing sections. See the CLUSTER EDIT Commands for a complete description.

ACCESS (__:__): This command will access a stored segment. The title of this segment is established as the input or active title.

Segment Creation:

BUILD (__:__): This command will cause a new segment to be built. It establishes the output title for any transformation for the ADD X,Y,Z Command the the COPY Command.

COPY (__:__): This command will cause the data stored under the specified title to be copied to the title specified by the BUILD Command.

Data Display:

LIST (__:__): This command will cause a listing of the specified segment to be printed.

Miscellaneous Commands:

MENU

OMIT

EXIT

These are the general utility commands and have the same meaning as described in CLUSTER EDIT.

Program Output

Since the GTM program is basically a geometry manipulation tool and highly interactive, program output essentially remains transparent to the user. They are, however, two types of output which are applicable. The first is those which can be classified as geometric analysis outputs. The second is formatted geometry.

Geometric analysis output includes mass properties evaluations, program response to user input, geometric displays and geometric parameters. The formatted geometry output is the cornerpoint geometry sets used by other technology modules.
SPECIAL USER INSTRUCTIONS

The GTM operates upon a data base of stored information which is manipulated by a language structure which is the primary means for the user to input to the program. These language elements are stored in the data base as character strings as well as the menus which provide the user with a summary of commands for the particular language set during execution.

To become well versed in the language and subsequent capabilities that can be obtained in GTM, the user must first become acquainted with the most commonly used functions of the GTM. Once he becomes familiar with these common functions, he will be able to graduate to the higher level functions.

To provide guidance in the use of the GTM, a set of instructions are provided to familiarize the user in the commonly used functions. In addition, a training aid in the form of a summary of GTM commands, command options and descriptions is provided in Appendix A. Sample cases are also given in Appendixes C and D.

How to Execute GTM

Operating under the Exec 8 system, the following sequence of commands will provide access to GTM:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>@RUN ID,ACCT.NO., ORGANIZATION</td>
<td>Exec 8 Run Card</td>
</tr>
<tr>
<td>@ASG,T 1.,F111500</td>
<td>Assigns temporary file 1 for the data base.</td>
</tr>
<tr>
<td>@COPY DATA5.,1.</td>
<td>Copies the data base on file DATA5 to temporary file 1. See Appendix B for the current contents of the GTM DATA BASE.</td>
</tr>
<tr>
<td>@XQT GTM2.GTM</td>
<td>Executes GTM, the program will now require user input.</td>
</tr>
</tbody>
</table>

How to Call a Menu

SAMPLE CASE 1: A menu of available commands may be obtained at any time by commanding the following:
A menu of available commands for the particular language set will be generated.

How to Access Stored Geometry

SAMPLE CASE 2:

COMMAND

DESCRIPTION

ACCESS CLUSTER S0147B Will access the cluster named S0147B.

ACCESS COMPONENT WING Will access the component named Wing of Cluster S0147B.

ACCESS SECTION SECTION 2 Will access the second section of component Wing of Cluster S0147B.

How to Input Externally Generated Gentry Geometry

SAMPLE CASE 3:

COMMAND

DESCRIPTION

INPUT Will provide access to the input module.

GENTRY GEOMETRY Specifies the format.

BCD FILE _n Specifies the file in which the external geometry is stored on.

NAME1 NAME2 User specifies geometry name, e.g., RED1, WHITE2, BLUE3.

NAMEn

EXIT Provides exit to master module.

How to Output Geometry

SAMPLE CASE 4:

COMMAND

DESCRIPTION

CLUSTER EDIT Provides access to cluster edit module.
SAMPLE CASE 4 (Continued)

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAGE OUTPUT SO147B</td>
<td>Will output the SO147B geometry in Gentry format to File 3.</td>
</tr>
</tbody>
</table>

How to Rotate Geometry

SAMPLE CASE 5:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER EDIT</td>
<td>Provides access to cluster edit module.</td>
</tr>
<tr>
<td>PITCH=40.</td>
<td>Specifies pitch angle of 40 deg.</td>
</tr>
<tr>
<td>YAW=-10.1</td>
<td>Specifies yaw angle of -10.1 degrees.</td>
</tr>
<tr>
<td>ROLL=.05</td>
<td>Specifies roll angle of .05 deg.</td>
</tr>
<tr>
<td>ACCESS CLUSTER SO147B</td>
<td>Access SO147B geometry.</td>
</tr>
<tr>
<td>ROT</td>
<td>Commands the rotation on the SO147B geometry.</td>
</tr>
</tbody>
</table>

How to Translate Geometry

SAMPLE CASE 6:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTER EDIT</td>
<td>Provides access to the cluster edit module.</td>
</tr>
<tr>
<td>XMOVE=10.</td>
<td>Specifies a X-translation of 10 inches.</td>
</tr>
<tr>
<td>YMOVE=-1500.</td>
<td>Specifies a Y-translation of -1500 inches.</td>
</tr>
<tr>
<td>ZMOVE=.001</td>
<td>Specifies a Z-translation of .001 inches.</td>
</tr>
<tr>
<td>ACCESS CLUSTER SO147B</td>
<td>Provides access to SO147B geometry.</td>
</tr>
<tr>
<td>ACCESS COMPONENT WING</td>
<td>Provides access to SO147B wing (only) geometry.</td>
</tr>
<tr>
<td>MOVE</td>
<td>Commands the move.</td>
</tr>
</tbody>
</table>
How to Display Geometry

SAMPLE CASE 7:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY S0147B</td>
<td>Specifies the display of S0147B geometry.</td>
</tr>
</tbody>
</table>

CONCLUDING REMARKS

The GTM has advanced to the state where geometry can be stored by name at several hierarchical levels in a tree structure. Editing of the data can be performed at all levels of data definition. The program contains most of the basic utilities which permit the development of user oriented manipulation of geometry and critical manipulation functions, such as scaling, rotation and translation has been programmed. The program is designed to be used in the demand or batch mode. Input is based on an easily expandable language structure input to the program. Additional user capability and utility functions can be added. The characteristics and capabilities of the GTM enable an engineer to:

1. Generate interior and exterior configuration geometry by inputting data from a remote terminal through keyboard input and selection from a data base of stored three dimensional geometrical shapes and standard section data.

2. Perturb configuration geometry by scaling, translation, rotation and mathematical mapping of component geometry.

3. Perturb the configuration by specifying mathematical surface fit functions.

4. Analyze the interior arrangement of geometric components with respect to each other and with respect to the external configuration to avoid volumetric conflict and violation of exterior lines of the vehicle.

5. Select components for the purpose of displaying three dimensional internal and external geometry.
6. Select components for the purpose of mass properties evaluation including weight, volume and center of gravity calculations.

7. Generate the required geometric interface data to be input into the data base which would be consistent with geometry data required for sizing, aerodynamics analysis packaging, weight and balance and structural analysis.

The GTM is a demand or batch program with standard Tektronix 4012 interface. The GTM will be easily adaptable to the interactive hardware and software systems anticipated in late 1975.
REFERENCES


The GTM Master Module Menu Index is a comprehensive list of available functions within the module. The Master Module is the point from which all submodules within the GTM can be accessed. In addition, it contains several commands which perform specific operations for data base and internal program management.

The following describes the functions and commands available within the master module:

**Primary Command:** IMAGE INPUT
**Optional Command:** GENTRY INPUT
**Description:** A user entry of "IMAGE INPUT" will access the image input module and its associated functions.

**Primary Command:** CLUSTER EDIT
**Optional Command:** SECTION EDIT
**Description:** A user entry of "CLUSTER EDIT" will access the CLUSTER Edit Module and its associated functions.

**Primary Command:** SEGMENT EDIT
**Optional Command:** NA
**Description:** A user entry of "SEGMENT EDIT" will access the segment edit module and its associated functions.

**Primary Command:** INPUT
**Optional Command:** NA
**Description:** A user entry of "INPUT" will access the input module and its associated functions.

**Primary Command:** CALCUL
**Optional Command:** NA
**Description:** A user entry of "CALCUL" will access the calculator module and its associated functions.

**Primary Command:** MENU
**Optional Command:** NA
**Description:** A user entry of "MENU" will generate a listing of the available commands and functions of the master module.
<table>
<thead>
<tr>
<th>Primary Command</th>
<th>Optional Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STOP</td>
<td>NA</td>
<td>A user entry of &quot;STOP&quot; will terminate all GTM activity.</td>
</tr>
<tr>
<td>SAVE DATA BASE</td>
<td>NA</td>
<td>A user entry of &quot;SAVE DATA BASE&quot; will store and save all geometry files in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the GTM data base for the current execution.</td>
</tr>
<tr>
<td>OPS STACK name:name:name:name:</td>
<td>NA</td>
<td>A user entry of &quot;OPS STACK name:name:name:name,&quot; will access the operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stack stored in the GTM data base under the specified name:name:name:name,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and perform the commands or functions associated within it.</td>
</tr>
<tr>
<td>EXTERNAL cc=name:name:name:OS=name:name:name:</td>
<td>NA</td>
<td>A user entry of &quot;EXTERNAL cc= name:name:name:OS=name:name:name:&quot; transfers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>control from the GTM to an external control card stack in the data base.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When GTM is reentered, the control stack specified by OS=name:name:name:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>will be executed.</td>
</tr>
<tr>
<td>MIMIC</td>
<td>NA</td>
<td>This command is mainly used in conjunction with &quot;canned&quot; operations. A user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>entry of &quot;MIMIC&quot; will display the user input for each computer inquiry.</td>
</tr>
<tr>
<td>HUSH</td>
<td>NA</td>
<td>A user entry of &quot;HUSH&quot; will negate the MIMIC command and disallow the display</td>
</tr>
</tbody>
</table>
|                                       |                  | of user inputs to computer inquiries. This command is mainly used in conjunc-
|                                       |                  | tion with "canned" operations.                                             |
The basic function of this module allows the user to select and command an input, in Gentry format, to be loaded from the data base and/or a specified unit. The data to be loaded can be specified in either BCD (alpha numeric) or binary (numeric) Gentry format.

The following describes the functions and commands available within the IMAGE INPUT MODULE.

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>DATA BASE name:name:name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;DATA BASE name:name:name&quot; will load from the data base the Gentry file under the specified name:name:name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>BINARY FILE UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>In response to a computer inquiry for the unit to be read, a user entry of &quot;BINARY FILE UNIT&quot; will specify the named binary file to be read.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>BCD FILE UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>In response to a computer inquiry for the unit to be read, a user entry of &quot;BCD FILE UNIT&quot; will specify the named BCD file to be read.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>MENU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;MENU&quot; will generate a listing of the available commands and functions of the Gentry input module.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>EXIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>FINISHED</td>
</tr>
<tr>
<td></td>
<td>INPUT COMPLETE</td>
</tr>
<tr>
<td></td>
<td>INPUT COMPLETED</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;EXIT&quot; or any of its options will cause an exit from the Gentry module to the master module.</td>
</tr>
</tbody>
</table>
Primary Command: CLUSTER name: name: name
Optional Command: CLUSTER NAME name: name: name
CL name: name: name
Description: A user entry of "CLUSTER name: name: name" in response to a GTM inquiry for a cluster name will assign the input name: name: name to the geometry.

Primary Command: COMPONENT name: name: name
Optional Command: COMPONENT NAME name: name: name
COM name: name: name
Description: A user entry of "COMPONENT name: name: name" in response to a GTM inquiry for a component name will assign the input name: name: name to the geometry.

Primary Command: DEFAULT
Optional Command: CL DEFAULT
CLUSTER DEFAULT
Description: A user entry of "DEFAULT" in response to a computer inquiry for a vehicle name will assign default names for the vehicle and its components. The default names are CLUSTER1, CLUSTER2 .......CLUSTERn.
The Cluster Edit Module allows a user to manipulate and display geometry at the cluster/component level. The basic functions of the module include geometric rotations and translations, cluster section editing and display. In addition, several options such as the calculation of volume and mass properties are provided by this module.

The following figure illustrates the basic menu structure of the Cluster Edit Module:

```
CLUSTER EDIT MODULE

GENERAL MENU

EDIT MENU  ACCESS MENU  COPY MENU  GEOMETRIC MANIPULATION MENU  DISPLAY MENU
```

The following describe the functions and commands associated with the general MENU:

- **Primary Command:** MENU
  - **Optional Command:** NA
  - **Description:** A user entry of "MENU" will provide a listing of the commands and functions of the cluster edit module.

- **Primary Command:** OMIT
  - **Optional Command:** NA
  - **Description:** A user entry of "OMIT" will exit the cluster edit module to the master module.
<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>Optional Command:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MENU EDIT</td>
<td>NA</td>
<td>A user entry of &quot;MENU EDIT&quot; will provide a listing of the EDIT MENU.</td>
</tr>
<tr>
<td>MENU ACCESS</td>
<td>NA</td>
<td>A user entry of &quot;MENU ACCESS&quot; will provide a listing of the ACCESS MENU.</td>
</tr>
<tr>
<td>MENU COPY</td>
<td>NA</td>
<td>A user entry of &quot;MENU COPY&quot; will provide a listing of the COPY MENU.</td>
</tr>
<tr>
<td>MENU GEOMETRIC MANIPULATION</td>
<td>NA</td>
<td>A user entry of &quot;MENU GEOMETRIC MANIPULATION&quot; will generate a listing of the MANIPULATION MENU.</td>
</tr>
<tr>
<td>MENU DISPLAY</td>
<td>NA</td>
<td>A user entry of &quot;MENU DISPLAY&quot; will generate a listing of the MENU DISPLAY.</td>
</tr>
<tr>
<td>EXIT</td>
<td>NA</td>
<td>A user entry of &quot;EXIT&quot; will terminate cluster edit activity and return to the master module.</td>
</tr>
</tbody>
</table>

The following describes the functions and commands associated with the EDIT MENU.

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>Optional Command:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUILD CLUSTER</td>
<td>BCL</td>
<td>A user entry of &quot;BUILD CLUSTER <em>name: name</em> <em>name: name</em> <em>name: name</em> &quot; will create a cluster with the specified name.</td>
</tr>
<tr>
<td>BUILD COMPONENT</td>
<td>B COML</td>
<td>A user entry of &quot;BUILD COMPONENT <em>name: name</em> <em>name: name</em> <em>name: name</em> &quot; will create a component with the specified name.</td>
</tr>
<tr>
<td>Primary Command:</td>
<td>BUILD SECTION  name: name: name</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>Optional Command:</td>
<td>B SEC  name: name: name</td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;BUILD SECTION  name: name: name&quot; will create a section with the specified name.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>BUILD SEGMENT  name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>B SEG  name: name: name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;BUILD SEGMENT  name: name: name&quot; will create a segment with the specified name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>INSERT CLUSTER  name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>I CL  name: name: name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;INSERT CLUSTER  name: name: name&quot; will insert the named cluster in the currently accessed cluster.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>INSERT SECTION  name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>I SEC  name: name: name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;INSERT SECTION  name: name: name&quot; will insert the named section ahead of the currently accessed section.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>INSERT SEGMENT  name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>I SEG  name: name: name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;INSERT SEGMENT  name: name: name&quot; will insert the named segment ahead of the currently accessed segment.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>INSERT COMPONENT  name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>I COM  name: name: name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;INSERT COMPONENT  name: name: name&quot; will insert the named component ahead of the currently accessed component.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>REPLACE SEGMENT  name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>R SEG  name: name: name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;REPLACE SEGMENT  name: name: name&quot; will replace the currently accessed segment with the specified name.</td>
</tr>
</tbody>
</table>
Primary Command: REPLACE COMPONENT \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: R COM \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "REPLACE COMPONENT \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will replace the currently accessed component with the specified name.

Primary Command: REPLACE CLUSTER \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: R CL \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "REPLACE CLUSTER \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will replace the currently specified cluster with the specified name.

Primary Command: REPLACE SECTION \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: R SEC \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "REPLACE SECTION \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will replace the currently accessed section with the specified name.

Primary Command: DELETE SEGMENT \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: D SEG \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "DELETE SEGMENT \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will delete the named segment from the currently accessed section.

Primary Command: DELETE SECTION \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: D SEC \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "DELETE SECTION \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will delete the named section from the currently accessed component.

Primary Command: REPLACE COMPONENT \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: D COM \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "DELETE COMPONENT \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will delete the named component from the currently accessed cluster.

Primary Command: DELETE CLUSTER \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Optional Command: D CL \_\_\_\_\_\_ name: \_\_\_\_\_\_ name
Description: A user entry of "DELETE CLUSTER \_\_\_\_\_\_ name: \_\_\_\_\_\_ name" will delete the named cluster from the list of available clusters.
Primary Command: TREELIST name: name: name
Optional Command: TL name: name: name
Description: A user entry of "TREELIST name: name: name" will provide a listing of the named cluster components and sections.

Primary Command: LIST AVAILABLE name: name: name
Optional Command: LACL
Description: A user entry of "LACL" will provide a listing of the available data base clusters.

Primary Command: LIST CLUSTER name: name: name
Optional Command: L CL name: name: name
Description: A user entry of "LIST CLUSTER name: name: name" will provide a listing of the components and sections within the named cluster.

Primary Command: LIST COMPONENT name: name: name
Optional Command: L COM name: name: name
Description: A user entry of "LIST COMPONENT name: name: name" will provide a listing of the sections within the named component.

Primary Command: LIST SECTION name: name: name
Optional Command: L SEC name: name: name
Description: A user entry of "LIST SECTION name: name: name" will generate a listing of the segments within the accessed section.

Primary Command: LIST SEGMENT name: name: name
Optional Command: L SEG name: name: name
Description: A user entry of "LIST SEGMENT name: name: name" will generate a listing of the named segment data.

The following describes the functions and commands associated with the ACCESS MENU:

Primary Command: ACCESS CLUSTER name: name: name
Optional Command: AC CL name: name: name
Description: A user entry of "ACCESS CLUSTER name: name: name" will access the named cluster geometry from the data base.
Primary Command: ACCESS COMPONENT name: name: name
Optional Command: AC COM name: name: name
Description: A user entry of "ACCESS COMPONENT name: name: name" will access the named component geometry from the data base.

Primary Command: START X : Y : Z
Optional Command: NA
Description: A user entry of "START X : Y : Z" initializes a X, Y and Z start point in the geometry. This command is mainly used in conjunction with scaling and editing function of the module.

Primary Command: STOP X : Y : Z
Optional Command: NA
Description: A user entry of "STOP X : Y : Z" initializes a X, Y and Z stop point in the geometry. This command is mainly used in conjunction with scaling and editing functions of the module.

Primary Command: LOCATE COMPONENT name: name: name
Optional Command: LOC COM name: name: name
Description: A user entry of "LOCATE COMPONENT name: name: name" will access the named component geometry and store it in the locate register.

Primary Command: LOCATE CLUSTER name: name: name
Optional Command: LOC CL name: name: name
Description: A user entry of "LOCATE CLUSTER name: name: name" will access the named CLUSTER geometry and store it in the locate register.

Primary Command: ACCESS=LOCATE
Optional Command: A LOC
Description: A user entry of "ACCESS=LOCATE" sets the Access register equal to the LOCATE register.

Primary Command: ACCESS SECTION name: name: name
Optional Command: AC SEC name: name: name
Description: A user entry of "ACCESS SECTION name: name: name" will access the named section geometry from the data base.
<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>ACCESS SEGMENT _name:_name:_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>AC SEG _name:_name:_name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;ACCESS SEGMENT _name:_name:_name&quot; will access the named segment geometry from the data base.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>LOCATE SEGMENT _name:_name:_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>LOC SEC _name:_name:_name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;LOCATE SEGMENT _name:_name:_name&quot; will access the named segment geometry and store it in the locate register.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>LOCATE SECTION _name:_name:_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>LOC SEC _name:_name:_name</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;LOCATE SECTION _name:_name:_name&quot; will access the named section geometry and store it in the locate register.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>LOCATE=ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>LOC=A</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;LOCATE=ACCESS&quot; will set the locate register equal to the access register.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>ZERO LOCATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>ZERO LOC</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;ZERO LOCATE&quot; will clear the locate register.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>ZERO ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>ZERO AC</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;ZERO ACCESS&quot; will clear the access register.</td>
</tr>
</tbody>
</table>

The following describes the functions and commands associated with the COPY MENU:

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>START _X:_Y:_Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;START _X:_Y:_Z&quot; specifies a start point at the named X,Y,Z position.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>STOP _X:_Y:_Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;STOP _X:_Y:_Z&quot; specifies a stop point at the named X,Y,Z position.</td>
</tr>
<tr>
<td>Primary Command:</td>
<td>Description:</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>COPY CLUSTER <em>name:</em> name</td>
<td>A user entry of &quot;COPY CLUSTER <em>name:</em> name&quot; will copy the component and sections of the accessed CLUSTER to the specified name.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>C CL <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>COPY COMPONENT <em>name:</em> name</td>
<td>A user entry of &quot;COPY COMPONENT <em>name:</em> name&quot; will copy the sections of the accessed component to the specified component name.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>C COM <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>COPY SECTION <em>name:</em> name</td>
<td>A user entry of &quot;COPY SECTION <em>name:</em> name&quot; will copy the segments of the accessed section to the named section.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>C SEC <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>COPY SEGMENT <em>name:</em> name</td>
<td>A user entry of &quot;COPY SEGMENT <em>name:</em> name&quot; will copy the accessed segments to the named segment.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>C SEG <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ADD SECTION <em>name:</em> name</td>
<td>A user entry of &quot;ADD SECTION <em>name:</em> name&quot; will copy the accessed data pointer to the named component.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>A SEC <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ADD SEGMENT <em>name:</em> name</td>
<td>A user entry of &quot;ADD SEGMENT <em>name:</em> name&quot; will copy the accessed data pointer to the named section.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>A SEG <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ADD COMPONENT <em>name:</em> name</td>
<td>A user entry of &quot;ADD COMPONENT <em>name:</em> name&quot; will copy the accessed data pointer to the named cluster.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>A COM <em>name:</em> name</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ADD CLUSTER <em>name:</em> name</td>
<td>A user entry of &quot;ADD CLUSTER <em>name:</em> name&quot; will copy all accessed component pointers to the named cluster.</td>
</tr>
<tr>
<td>Optional Command:</td>
<td>A CL <em>name:</em> name</td>
</tr>
</tbody>
</table>
Primary Command: IMAGE OUTPUT _name: name: name:
Optional Command: IMO _name: name: name:
Description: A user entry of "IMAGE OUTPUT _name: 
_name: name" will output the named geometry on temporary file 3.

The following describes the functions and commands associated with the Manipulation MENU.

Primary Command: PSI=value
Optional Command: YAW= value
Description: A user entry of "PSI=value" specifies a yaw rotation for the accessed geometry. This command is mainly used in conjunction with geometric rotations and displays.

Primary Command: THETA=value
Optional Command: PITCH=value
Description: A user entry of "THETA=value" specifies a pitch rotation for the accessed geometry. This command is mainly used in conjunction with geometric rotations and displays.

Primary Command: PHI=value
Optional Command: ROLL=value
Description: A user entry of "PHI=value" specifies a roll rotation for the accessed geometry. This command is mainly used in conjunction with geometric rotations and displays.

Primary Command: ROTATION CENTER _X_ : _Y_ : _Z_
Optional Command: RCEN _X_ : _Y_ : _Z_
Description: A user entry of "RCEN _X_ : _Y_ : _Z_ " specifies a X, Y and Z geometric center. This command is used exclusively for geometric rotations.

Primary Command: ROT
Optional Command: ROTATE
Description: A user entry of "ROT" will command a geometric rotation as specified previously by the PHI, PSI, THETA and RCEN commands.
<table>
<thead>
<tr>
<th>Primary Command</th>
<th>Optional Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMOVE=value</td>
<td>NA</td>
<td>A user entry of &quot;XMOVE=value&quot; will translate the geometry X station values by the amount specified once the move command is given.</td>
</tr>
<tr>
<td>YMOVE=value</td>
<td>NA</td>
<td>A user entry of &quot;YMOVE=value&quot; will translate the geometry Y station values by the amount specified once the move command is given.</td>
</tr>
<tr>
<td>ZMOVE=value</td>
<td>NA</td>
<td>A user entry of &quot;ZMOVE=value&quot; will translate the geometry Z station values by the amount specified once the move command is given.</td>
</tr>
<tr>
<td>MOVE</td>
<td>NA</td>
<td>A user entry of &quot;MOVE&quot; will command the program to execute the moves previously specified by XMOVE, YMOVE or ZMOVE.</td>
</tr>
<tr>
<td>XMAG=value</td>
<td>NA</td>
<td>A user entry of &quot;XMAG=value&quot; specifies a X station magnification factor. This entry is used in conjunction with the scale command.</td>
</tr>
<tr>
<td>YMAG=value</td>
<td>NA</td>
<td>A user entry of &quot;YMAG=value&quot; specifies a Y station magnification factor. This entry is used in conjunction with the scale command.</td>
</tr>
<tr>
<td>ZMAG=value</td>
<td>NA</td>
<td>A user entry of &quot;ZMAG=value&quot; specifies a Z station magnification factor. This entry is used in conjunction with the scale command.</td>
</tr>
</tbody>
</table>
Primary Command: MAG=value
Optional Command: NA
Description: A user entry of "MAG=value" will set the XMAG, YMAG and ZMAG values equal to the specified value.

Primary Command: MAGNIFICATION CENTER X Y Z
Optional Command: MCEN X Y Z
Description: A user entry of "MAGNIFICATION CENTER X Y Z" will set the magnification center to the specified values.

Primary Command: SCALE
Optional Command: SC
Description: A user entry of "SCALE" will command the magnifications specified previously by MCEN, XMAG, YMAG and ZMAG.

Primary Command: RHO=value
Optional Command: NA
Description: A user entry of "RHO=value" specifies the density for a volume and mass property report.

Primary Command: H=value
Optional Command: NA
Description: A user entry of "H=value" specifies a wall thickness for a volume and mass property report.

Primary Command: VAMP name: name: name
Optional Command: VOLUME name : name: name
Description: A user entry of "VAMP name: name: name" will generate a volume and mass property report for the named geometry.

The following describes the functions and commands associated with the DISPLAY MENU:

Primary Command: DISPLAY--name: name: name
Optional Command: DIS--name: name: name
Description: A user entry of "DISPLAY--name: name: name" will display the named geometry and disregard any previously calculated scaling factors.
Primary Command: DISPLAY name: name: name
Optional Command: DIS name: name: name
Description: A user entry of "DISPLAY name: name: name" will generate a display of the named geometry.

Primary Command: DISPLAY+ name: name: name
Optional Command: DIS+ name: name: name
Description: A user entry of "DISPLAY+ name: name: name" will generate an overlay display of the named geometry.

Primary Command: RFILT=value
Optional Command: RESOLUTION FILTER=value
Description: A user entry of "RFILT=value" will suppress a vector from being plotted that is greater than that specified.

Primary Command: ZOOM factor X Y
Optional Command: NA
Description: A user entry of "ZOOM factor X Y" specifies a zoom with a magnification factor, X screen and Y screen position.

Primary Command: REFRESH
Optional Command: NA
Description: A user entry of "REFRESH" will erase any current display images and redisplay all the previous images.

Primary Command: AFILT=value
Optional Command: AREA FILTER=value
Description: A user entry of "AFILT=value" specifies a limit area in which any values greater than that specified will not be plotted.

Primary Command: HL
Optional Command: NA
Description: A user entry of "HL" activates the hidden line algorithm. This command is used in conjunction with the geometric display option.

Primary Command: NOHL
Optional Command: NA
Description: A user entry of "NOHL" negotiates the hidden line algorithm. This command is used in conjunction with the geometric display option.
Primary Command: SYM
Optional Command: NA
Description: A user entry of "SYM" will generate symmetric displays when used in conjunction with the display option.

Primary Command: NOSYM
Optional Command: NA
Description: A user entry of "NOSYM" negotiates the SYM command.

Primary Command: PICTURE SIZE X Y
Optional Command: PICSIZ X Y
Description: A user entry of "PICTURE SIZE X Y" will generate a screen window specified by X and Y (inches).
The basic function of the SEGMENT EDIT module allows the user to manipulate geometry at the segment level. The manipulation capability includes rotation, translations, scaling, creation and display of segment geometry.

The following describes the general operations of the SEGMENT EDIT module:

**Primary Command:** COPY SEGMENT _name:_ name
**Optional Command:** COPY _name:_ name
**Description:** A user entry of "COPY SEGMENT _name:_ name" will copy the accessed segment geometry to the specified name.

**Primary Command:** BUILD SEGMENT _name:_ name
**Optional Command:** BUILD _name:_ name
**Description:** A user entry of "BUILD SEGMENT _name:_ name" will create a segment by the name specified.

**Primary Command:** ACCESS CLUSTER _name:_ name
**Optional Command:** AC CL _name:_ name
**Description:** A user entry of "ACCESS CLUSTER _name:_ name" will access the named CLUSTER geometry.

**Primary Command:** ACCESS COMPONENT _name:_ name
**Optional Command:** AC COM _name:_ name
**Description:** A user entry of "ACCESS COMPONENT _name:_ name" will access the named component geometry.

**Primary Command:** ACCESS SECTION _name:_ name
**Optional Command:** AC SEC _name:_ name
**Description:** A user entry of "ACCESS SECTION _name:_ name" will access the named section geometry.

**Primary Command:** ACCESS _name:_ name
**Optional Command:** AC _name:_ name
**Description:** A user entry of "ACCESS _name:_ name" will access the segment geometry specified by the entered name.
Primary Command: LIST SEGMENT _name: name: name
Optional Command: LIST _name: name: name
L _name: name: name
Description: A user entry of "LIST SEGMENT _name: name: name" will provide a listing of the named segment geometry.

Primary Command: DISPLAY SEGMENT _name: name: name
Optional Command: DISPLAY _name: name: name
DIS _name: name: name
Description: A user entry of "DISPLAY SEGMENT _name: name: name" will provide a three view of the named segment geometry.

Primary Command: MENU
Optional Command: NA
Description: A user entry of "MENU" will generate a listing of the available functions and commands within the segment edit module.

Primary Command: OMIT
Optional Command: EXIT
Description: A user entry of "OMIT" will terminate segment edit activity and exit control to the master module.

Primary Command: COPY SECTION _name: name: name
Optional Command: C SEC _name: name: name
Description: A user entry of "COPY SECTION _name: name: name" will copy the accessed segment to the named section.

The following describes the POINT EDIT commands associated with the SEGMENT EDIT module:

Primary Command: FIND X Y Z
Optional Command: FIND POINT X Y Z
Description: A user entry of "FIND X Y Z" will locate the named X,Y,Z point within the segment. The FIND command will locate the point nearest to the specified values of X,Y,Z.

Primary Command: ADD POINT X Y Z
Optional Command: ADD X Y Z
A X Y Z
Description: A user entry of "ADD POINT X Y Z" will add to the existing segment geometry the specified X, Y and Z value.
Primary Command: REPLACE POINT X Y Z
Optional Command: REPLACE X Y Z
R X Y Z
Description: A user entry of "REPLACE POINT X Y Z" will replace the accessed point with the specified X Y and Z value.

Primary Command: DELETE POINT X Y Z
Optional Command: DELETE X Y Z
D X Y Z
Description: A user entry of "DELETE POINT X Y Z" will delete the specified X, Y, Z point from the accessed segment geometry.

Primary Command: INSERT POINT X Y Z
Optional Command: INSERT X Y Z
I X Y Z
Description: A user entry of "INSERT POINT X Y Z" will insert the specified X, Y, Z values in the accessed segment.

Primary Command: DEFINE POINT X Y Z
Optional Command: DEFINE X Y Z
DEF X Y Z
Description: A user entry of "DEFINE POINT X Y Z" will insert the specified X, Y, Z values in the access register.

The following describes the Segment Operations:

Primary Command: START X Y Z
Optional Command: NA
Description: A user entry of "START X Y Z" will specify a start point as per X, Y and Z entry. This command is used in conjunction with the operational Segment command.

Primary Command: STOP X Y Z
Optional Command: NA
Description: A user entry of "STOP X Y Z" will specify a stop point as per X, Y and Z entry. This command is used in conjunction with the operational Segment command.
Primary Command: NSTART=value
Optional Command: NA
Description: A user entry of "NSTART=value" will establish a start boundary with the specified value.

Primary Command: NSTOP=value
Optional Command: NA
Description: A user entry of "STOP=value" will establish a stop boundary with the specified value.

Primary Command: NSEG=value
Optional Command: NA
Description: A user entry of "NSEG=value" will establish a segment boundary with the specified value. Used in conjunction with the EQARC command.

Primary Command: EQARC name: name: name
Optional Command: EQLEN name: name: name
Description: A user entry of "EQARC name: name: name" will specify the segment to be equally divided into N segments.

Primary Command: XMOVE=value
Optional Command: NA
Description: A user entry of "XMOVE=value" specifies the X value for a translation of the segment geometry. This command is used in conjunction with the MOVE command.

Primary Command: YMOVE=value
Optional Command: NA
Description: A user entry of "YMOVE=value" specifies the Y value for a translation of the segment geometry. This command is used in conjunction with the MOVE command.

Primary Command: ZMOVE=value
Optional Command: NA
Description: A user entry of "ZMOVE=value" specifies the Z value for a translation of the segment geometry. This command is used in conjunction with the MOVE command.
Primary Command: MOVE
Optional Command: NA
Description: A user entry of "MOVE" will perform the translations specified by XMOVE, YMOVE and ZMOVE.

Primary Command: PSI=value (Deg)
Optional Command: YAW=value (Deg)
Description: A user entry of "PSI=value" specifies a YAW value for a rotation of the segment geometry. This command is used in conjunction with the ROT command.

Primary Command: THETA=value (Deg)
Optional Command: PITCH=value (Deg)
Description: A user entry of "THETA=value" specifies a pitch value for rotation of the segment geometry. This command is used in conjunction with the ROT command.

Primary Command: PHI=value (Deg)
Optional Command: ROLL=value (Deg)
Description: A user entry of "PHI=value" specifies a roll value for rotation of the segment geometry. This command is used in conjunction with the ROT command.

Primary Command: ROTATION CENTER X Y Z
Optional Command: RCEN X Y Z
Description: A user entry of "RCEN X Y Z" describes a rotation center with the specified X,Y,Z values.

Primary Command: ROTATE
Optional Command: ROT
Description: A user entry of "ROTATE" will perform the rotations specified by PHI, THETA and PSI.

Primary Command: XMAG=value
Optional Command: NA
Description: A user entry of "XMAG=value" specifies an X magnification factor. Used in conjunction with the scale command.

Primary Command: YMAG=value
Optional Command: NA
Description: A user entry of "YMAG=value" specifies a Y magnification factor. Used in conjunction with the scale command.
Primary Command: MCEN X Y Z
Optional Command: MAGNIFICATION CENTER X Y Z
Description: A user entry of "MCEN X Y Z" specifies a magnification center with the named X, Y and Z values. This command is used in conjunction with the scale command.

Primary Command: MAG=value
Optional Command: NA
Description: A user entry of "MAG=value" will set XMAG, YMAG and ZMAG equal to the specified value.

Primary Command: ZMAG=value
Optional Command: NA
Description: A user entry of "ZMAG=value" specifies a Z magnification factor. Used in conjunction with the scale command.

Primary Command: SCALE
Optional Command: SC
Description: A user entry of "SCALE" will perform the magnification specified previously by MCEN, XMAG, YMAG and ZMAG.

Primary Command: PLANE X Y Z PSI THETA PHI
Optional Command: NA
Description: A user entry of "PLANE X Y Z PSI THETA PHI" defines a plane with the X,Y,Z coordinates and Pitch, Roll and Yaw rotation specified.

Primary Command: XCUT=value
Optional Command: NA
Description: A user entry of "XCUT=value" defines a Y-Z plane located at the specified X station.

Primary Command: YCUT=value
Optional Command: NA
Description: A user entry of "YCUT=value" defines a X-Z plane located at the specified Y station.
Primary Command: ZCUT=value
Optional Command: NA
Description: A user entry of "ZCUT=value" defines an X-Y plane located at the specified Z station.

Primary Command: PCUT
Optional Command: NA
Description: A user entry of "PCUT" specifies a cut positive to the normal plane as previously specified by PLANE, XCUT, YCUT and ZCUT.

Primary Command: MCUT
Optional Command: NA
Description: A user entry of "MCUT" specifies a cut negative to the normal plane as previously specified by PLANE, XCUT, YCUT and ZCUT.
The basic function of the calculator module provides the user with a series of basic calculator capabilities.

The following describes the functions and commands of the CALCULATOR Module:

**Primary Command: ENTER=value**
**Optional Command: E=value**
**Description:** A user entry of "ENTER=value" will enter the specified number.

**Primary Command: PLUS**
**Optional Command: ADD**
**Description:** A user entry of "PLUS" will perform an addition of the stored numbers.

**Primary Command: MINUS**
**Optional Command: SUB**
**Description:** A user entry of "MINUS" will perform a subtraction of the stored numbers.

**Primary Command: DIV**
**Optional Command: /**
**Description:** A user entry of "DIV" will perform a division of the stored numbers.

**Primary Command: MPY**
**Optional Command: ***
**Description:** A user entry of "MPY" will perform a multiplication of the stored numbers.

**Primary Command: COS**
**Optional Command: NA**
**Description:** A user entry of "COS" will take the COS of the stored number.

**Primary Command: SIN**
**Optional Command: NA**
**Description:** A user entry of "SIN" will take the SIN of the stored number.

**Primary Command: TAN**
**Optional Command: NA**
**Description:** A user entry of "TAN" will take the TANGENT of the stored number.
Primary Command: EXIT
Optional Command: NA
Description: A user entry of "EXIT" will terminate calculator activity and return to the master module.

Primary Command: MENU
Optional Command: NA
Description: A user entry of "MENU" will generate a listing of the available commands in the calculator module.

Primary Command: CLEAR
Optional Command: NA
Description: A user entry of "CLEAR" will clear (zero) any previously stored entry.
The basic functions of the INPUT module allows the user to enter data blocks and any type of information.

The following describes the commands available in the input module:

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>BCD INPUT _name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;BCD INPUT _name: name: name&quot; will store the card image data in the data base under the specified name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primary Command:</th>
<th>Numeric Input _name: name: name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional Command:</td>
<td>NA</td>
</tr>
<tr>
<td>Description:</td>
<td>A user entry of &quot;NUMERIC INPUT _name: name: name&quot; will enter numeric data under the specified name. The data is read in free field format.</td>
</tr>
</tbody>
</table>
APPENDIX B
GTM DATA BASE

The following provides a listing of the current GTM geometry data base contents on file EX42-00002.*DATA5:

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO147B</td>
<td>SHUTTLE ORBITER CLUSTER</td>
</tr>
<tr>
<td>NOSE</td>
<td>NOSE CLUSTER</td>
</tr>
<tr>
<td>TAIL</td>
<td>TAIL CLUSTER</td>
</tr>
<tr>
<td>OMSSUB</td>
<td>OMS SUBTRACTION SURFACE</td>
</tr>
<tr>
<td>OMSPOD</td>
<td>OMS POD</td>
</tr>
<tr>
<td>OWING</td>
<td>OUTER WING</td>
</tr>
<tr>
<td>IWING</td>
<td>INNER WING</td>
</tr>
<tr>
<td>BODY</td>
<td>BODY STRUCTURE</td>
</tr>
<tr>
<td>LE</td>
<td>WING LEADING EDGE</td>
</tr>
<tr>
<td>BFLAP</td>
<td>BODY FLAP</td>
</tr>
<tr>
<td>SRM</td>
<td>SOLID ROCKET BOOSTER</td>
</tr>
<tr>
<td>SHRDL</td>
<td>LOWER PAYLOAD SHROUD (LLV)</td>
</tr>
<tr>
<td>SHRDV</td>
<td>UPPER PAYLOAD SHROUD (LLV)</td>
</tr>
<tr>
<td>STAGE6</td>
<td>SIXTH STAGE LLV</td>
</tr>
<tr>
<td>STAGE5</td>
<td>FIFTH STAGE LLV</td>
</tr>
<tr>
<td>STAGE4</td>
<td>FOURTH STAGE LLV</td>
</tr>
<tr>
<td>STAGE3</td>
<td>THIRD STAGE LLV</td>
</tr>
<tr>
<td>STAGE2</td>
<td>SECOND STAGE LLV</td>
</tr>
<tr>
<td>STAGE1</td>
<td>FIRST STAGE LLV</td>
</tr>
<tr>
<td>LLV</td>
<td>LARGE LEFT VEHICLE</td>
</tr>
</tbody>
</table>
APPENDIX C - SAMPLE PROBLEM

AERODYNAMIC SURFACES

The following pages contain a sample input and graphical representation for the construction of a wing and vertical tail using the GTM and auxiliary programs. The wing and tail surfaces use standard NACA airfoils as follows:

- Wing Root: NACA 2408
- Wing Tip: NACA 2412
- Vertical Root and Tip: NACA 0008

The standard airfoil geometry was generated by the AIRFOIL Program of reference 2 and passed to the GTM on a specially formatted file. The graphics representation was generated by the IMAGE Program of reference 2.
INPUT
READ FILE 4
EXIT
SEGMENT EDIT

L NACA 2408 UPPER
  NSEG=10
  EQARC NACA 2408 UPPER
  NSEG=10
  EQARC NACA 2408 LOWER
  NSEG=10
  EQARC NACA 2412 UPPER
  NSEG=10
  EQARC NACA 2412 LOWER
  MAG=9.72
  SC NACA 2408 UPPER
  SC NACA 2408 LOWER
  YAW=180.
  ROT NACA 2408 UPPER
  ROT NACA 2408 LOWER
  XMOVE=-1704.
  YMOVE=272.
  ZMOVE=-132.
  MOVE NACA 2408 UPPER
  MOVE NACA 2408 LOWER
  MAG=2.01
  SC NACA 2412 UPPER
  SC NACA 2412 LOWER
  YAW=180.
  ROT NACA 2412 UPPER
  ROT NACA 2412 LOWER
  XMOVE=-2475.
  YMOVE=954.
  ZMOVE=-132.
  MOVE NACA 2412 UPPER
  MOVE NACA 2412 LOWER
EXIT

SECTION EDIT

BUILD VEHICLE_PARALLEL_BURN_SSTO
BUILD COMPONENT_WING_UPPER
BUILD SECTION_ROOT
COPY SEGMENT_NACA_2408_UPPER
BUILD SECTION_TIP
COPY SEGMENT_NACA_2412_UPPER
BUILD COMPONENT_WING_LOWER
BUILD SECTION_ROOT
COPY SEGMENT_NACA_2412_LOWER
BUILD SECTION_TIP
COPY SEGMENT_NACA_2412_LOWER
TREE LIST_PARALLEL_BURN_SSTO
EXIT

WING GENERATION INPUT.
SEGMENT_EDIT

ACCESS_NACA_0008_UPPER
BUILD_SEGMENT_TIP
COPY_SEGMENT
ACCESS_NACA_0008_UPPER
BUILD_SEGMENT_ROOT
COPY_SEGMENT
ACCESS_ROOT
NSEG = 10
EQARC
YAW = 180
ROLL = -90
ROLL = 90
ROT
MAG = 4.51
SC
XMOVE = -2222
ZMOVE = 147
MOVE
ACCESS_TIP
NSEG = 10
EQARC
ROT
MAG = 2.02
SC
XMOVE = -2660
ZMOVE = 609
MOVE
EXIT
SECTION_EDIT

ACCESS_VEHICLE_PARALLEL_BURN_SS10
BUILD = ACCESS
BUILD_COMPONENT_VIAIL
ADD_SEGMENT_TIP
ADD_SEGMENT_ROOT
TREE_LIST_PARALLEL_BURN_SS10
ACCESS_VEHICLE_PARALLEL_BURN_SS10
OUTPUT
EXIT
SAVE_DATA_BASE
STOP

VERTICAL TANK GENERATION INPUT:

ORIGINAL PAGE IS OF POOR QUALITY
AERODYNAMIC SURFACE OUTPUT.
GULL WING GEOMETRY GENERATION

Five steps are included which generate the geometry of a multiply deflected wing shape. Separate executions using EDIN were used to demonstrate the ability of the GTM to access and modify stored geometry as well as to create geometry. The results of each step are displayed. The five steps are:

1. Create a flat, cranked wing.
2. Vertical Stabilizer surface.
3. Flap surface.
4. Elevon and Rudder surfaces.
5. Rotations and deflections.

Step 1: Create a Flat, Cranked Wing

The cranked wing is created using two standard NACA airfoil sections generated by the AIRFOIL program of reference 2. These are used as follows:

<table>
<thead>
<tr>
<th>Section</th>
<th>Airfoil Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root</td>
<td>NACA 2408</td>
</tr>
<tr>
<td>Crank</td>
<td>NACA 2408</td>
</tr>
<tr>
<td>Tip</td>
<td>NACA 2412</td>
</tr>
</tbody>
</table>

These sections are scaled and rotated into the aircraft coordinate system and assembled into components using the following command sequence for the GTM:

```
```
*EXECUTE GTM*

C.

C. BUILD FLAT CRANK WING

C.

INPUT

C.

C. READ AIRFOIL INPUT SECTIONS

C. NOTE: OUTPUT FROM AIRFOIL

C.

NACA 2408 UPPER
C.

NACA 2408 LOWER
C.

NACA 2412 UPPER
C.

NACA 2412 LOWER
C.

NACA 0008 UPPER
C.

NACA 0008 LOWER

C.

READ FILE 4

EXIT

SEGMENT EDIT

C.

C. ROTATE AIRFOIL SECTIONS TO GET INTO AC COORDINATE SYSTEM.

C.

YAW = 180
ROT NACA 2408 UPPER
ROT NACA 2408 LOWER
ROT NACA 2412 UPPER
ROT NACA 2412 LOWER

C.

C. COPY TO DIFFERENTLY NAMED SECTIONS

C.

BUILD ROOTU
COPY NACA 2408 UPPER
BUILD ROOTL
COPY NACA 2408 LOWER
BUILD CRANKU
COPY NACA 2408 UPPER
BUILD CRANKL
COPY NACA 2408 LOWER

C.

C. REDUCE POINTS PER SEGMENT TO 12

C.

NSEG = 12
EQARC ROOTU
EQARC ROOTL
EQARC CRANKU
EQARC CRANKL
EQARC NACA 2412 UPPER
EQARC NACA 2412 LOWER

ORIGINAL PAGE IS OF POOR QUALITY
C.
SCALE A MOVE ROOT SEGMENTS
C.
   MAG = 14.206
   XMOVE = -1256
   YMOVE = 272
   ZMOVE = -132
   SC ROOTU
   MOVE
   SC ROOTL
   MOVE
C.
SCALE AND MOVE CRANK SEGMENTS
C.
   MAG = 4.51
   XMOVE = -2225
   YMOVE = 573
   SC CRANKU
   MOVE
   SC CRANKL
   MOVE
C.
SCALE AND MOVE TIP SEGMENTS
C.
   MAG = 2.01
   XMOVE = -2475
   YMOVE = 954
   SC NACA 2412 UPPER
   MOVE
   SC NACA 2412 LOWER
   MOVE
C.
SEGMENT MANIPULATIONS COMPLETE
C.
EXIT

ORIGINAL PAGE IS
OF POOR QUALITY.

GTM INPUT--FLAT, CRANKED WING (cont.)
SECTION EDIT
C. CREATE VEHICLE TREE
C.

BUILD VEHICLE ORBITER
BUILD COMPONENT WING UPPER
BUILD SECTION ROOT
COPY SEGMENT ROOTU
BUILD SECTION CRANK
COPY SEGMENT CRANKU
BUILD SECTION TIP
COPY SEGMENT NACA 2412 UPPER
BUILD COMPONENT WING LOWER
BUILD SECTION TIP
COPY SEGMENT NACA 2412 LOWER
BUILD SECTION CRANK
COPY SEGMENT CRANKL
BUILD SECTION ROOT
COPY SEGMENT ROOTL
ACCESS VEHICLE ORBITER
TREE LIST
C.
C. OUTPUT INFORMATION FILE TO IMAGE
C.

OUTPUT ORBITER
EXIT
C.
C. FLAT WING IS COMPLETED
C.
SAVE DATA BASE
STOP
*EOF

GTM INPUT--FLAT, CRANKED WING (cont.)
Step 2: Vertical Stabilizer Surface

The vertical stabilizer surface is generated from the following sections:

<table>
<thead>
<tr>
<th>Tip</th>
<th>NACA 0008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cap</td>
<td>Input section data</td>
</tr>
</tbody>
</table>

These sections are scaled and rotated and appended to the wing components. The following command sequence for the GTM is used for these operations:
EXECUTE GTM*

BUILD WING TIP/VTAIL SECTION--FLAT

SEGMENT EDIT

ROTATE AND SCALE TIP SEGMENTS

ACCESS NACA 0008 UPPER
NSEG = 12
EQARC
YAW = 180
MAG = 2.01
ROT
SC
ACCESS NACA 0008 LOWER
NSEG = 12
EQARC
ROT
SC

MOVE TIP SEGMENTS INTO DESIRED POSITION

XMOVE = -2475
YMOVE = 1054
ZMOVE = -132
MOVE NACA 0008 UPPER
MOVE NACA 0008 LOWER

BUILD TIP CAP SEGMENT

BUILD SEGMENT LINE
ADD 0,0,0
ADD -2.5,0
ADD -4.8,0
ADD -6.9,0
ADD -1,1,0

SCALE AND MOVE CAP SEGMENT

XMAG = 201
YMAG = 20.
ZMAG = 1.
SC
XMOVE = -2475
YMOVE = 1064
ZMOVE = -132
MOVE
NSEG = 12
EQARC
CREATE UNIQUE NAMES FOR SEGMENTS TO BE INSERTED

BUILD SEGMENT VT1
COPY NACA 0008 LOWER
BUILD SEGMENT VT2
COPY LINE

EXIT
SECTION EDIT

ADD TAIL SECTIONS TO EXISTING WING

ACCESS VEHICLE ORBITER
ACCESS COMPONENT WING UPPER

SET UP THE BUILD REGISTER

BUILD = ACCESS
BUILD SECTION VT1
COPY SEGMENT NACA 0008 UPPER
BUILD SECTION VT2
COPY SEGMENT LINE
ACCESS COMPONENT WING LOWER

LOWER TIP SECTIONS MUST BE INSERTED TO BE PLACED IN THE POSITION.

ACCESS SECTION TIP
INSERT SECTION VT1
ACCESS SECTION VT1
INSERT SECTION VT2
TREE LIST ORBITER
OUTPUT ORBITER

EXIT
SAVE DATA BASE
STOP

*EOF

GTM INPUT--VERTICAL STABILIZER SURFACE (cont.)
FLAT, CRANKED WING WITH VERTICAL STABILIZER ADDED

ORIGINAL PAGE IS OF POOR QUALITY
Step 3: Flap Surface

The Flap is created by cutting the Wing Root and Wing Crank sections with a plane. The portions forward of the cut become replacement wing sections and portions aft of the cut become sections for the Flap component. The following command sequence for the GTM is used for these operations:
EXECUTE GTM#
C.---------------------------------------------------------------
C. BUILD INNER WING FLAP
C.---------------------------------------------------------------
SECTION EDIT
C. FLAP WILL BUILD FROM ROOT AND CRANK SECTIONS. THESE SECTIONS
C. MUST BE COPIED TO SEGMENTS
C. ACCESS VEHICLE ORBITER
ACCESS COMPONENT WING UPPER
BUILD SEGMENT FLAPUR
COPY SECTION ROOT
BUILD SEGMENT FLAPUT
COPY SECTION CRANK
ACCESS COMPONENT WING LOWER
BUILD SEGMENT FLAPLR
COPY SECTION ROOT
BUILD SEGMENT FLAPLT
COPY SECTION CRANK
EXIT
SEGMENT EDIT
C. FLAP SEGMENTS WILL BE BUILT FROM THE NEGATIVE PORTIONS OF THE
C. SEGMENTS AS REFERENCED FROM THE CUTTING PLANE.
C. NEW WING SEGMENTS WILL BE THE POSITIVE PORTIONS.
C. XCUT = -2476
   BUILD SEGMENT ROOT UPPER
   PCUT FLAPUR
   ACCESS FLAPUR
   MCUT
   BUILD SEGMENT CRANKI UPPER
   PCUT FLAPUT
   ACCESS FLAPUT
   MCUT
   BUILD SEGMENT ROOT LOWER
   PCUT FLAPLR
   ACCESS FLAPLR
   MCUT
   BUILD SEGMENT CRANKI LOWER
   PCUT FLAPLT
   ACCESS FLAPLT
   MCUT
C. INSURE 12 POINTS FOR WING SEGMENTS
C. ACCESS ROOT UPPER
   NSEG = 12
   EQARC
ACCESS ROOT LOWER
   NSEG = 12
   EQARC

GTM INPUT--FLAP SURFACE 91
ACCESS CRANKI UPPER
NSEG = 12
EQARC
ACCESS CRANKI LOWER
NSEG = 12
EQARC

C. EXTINGUISH THE POINT DEFINITION FOR THE FLAP SEGMENTS
C.

ACCESS FLAPUR
NSEG = 1
EQARC
ACCESS FLAPLR
NSEG = 1
EQARC
ACCESS FLAPLT
NSEG = 1
EQARC
ACCESS FLAPUT
NSEG=1
EQARC

EXIT
SECTION EDIT
C. --------------
C.

C. BUILD FLAP TREE STRUCTURE
C.-

ACCESS VEHICLE ORBITER
BUILD = ACCESS
BUILD COMPONENT FLAP UPPER
BUILD SECTION ROOT
COPY SEGMENT FLAPUR
BUILD SECTION TIP
COPY SEGMENT FLAPUR
BUILD COMPONENT FLAP LOWER
BUILD SECTION TIP
COPY SEGMENT FLAPLT
BUILD SECTION ROOT
COPY SEGMENT FLAPLT
C.

C. CORRECT WING STRUCTURE TO ACCEPT THE FLAP
C.

ACCESS COMPONENT WING UPPER
ACCESS SECTION ROOT
REPLACE SECTION ROOT UPPER
ACCESS SECTION CRANK
INSERT SECTION CRANKI UPPER
ACCESS COMPONENT WING LOWER
ACCESS SECTION ROOT
REPLACE SECTION ROOT LOWER
ACCESS SECTION ROOT LOWER
INSERT SECTION CRANKI LOWER
TREE LIST ORBITER
OUTPUT ORBITER
EXIT
SAVE DATA BASE
STOP

*EOF

GTM INPUT--FLAP SURFACE (cont.)
Step 4: Elevon and Rudder Surfaces

The Elevon is created by cutting the wing components from the Crank sections to the Tip sections. The Rudder is created by cutting the Wing components from the Tip sections to the VT1 sections. The following command sequence for the GTM is used for these operations:
EXECUTE GTM#

C. BUILD ELEVON AND RUDDER

SECTION EDIT

C. ELEVON IS BUILT FROM CRANK AND TIP
C. RUDDER IS BUILT FROM ELEVON TIP AND VT1

ACCESS VEHICLE ORBITER
ACCESS COMPONENT WING UPPER
BUILD SEGMENT ELEVON UR
COPY SECTION CRANK
BUILD SEGMENT ELEVON UT
COPY SECTION TIP
BUILD SEGMENT RUDDER UT
COPY SECTION VT1
ACCESS COMPONENT WING LOWER
BUILD SEGMENT ELEVON LR
COPY SECTION CRANK
BUILD SEGMENT ELEVON LT
COPY SECTION TIP
BUILD SEGMENT RUDDER LT
COPY SECTION VT1

EXIT

SEGMENT EDIT

C. ELEVON AND RUDDER ARE ON THE NEGATIVE SIDE OF THE CUTTING PLANE
C. THE REDEFINED WING IS ON THE POSITIVE SIDE

Xcut = -2576
BUILD CRANKO UPPER
PCUT ELEVON UR
ACCESS ELEVON UR
MCUT
BUILD TIP UPPER
PCUT ELEVON UT
ACCESS ELEVON UT
MCUT
BUILD VT1 UPPER
PCUT RUDDER UR
ACCESS RUDDER UR
MCUT
BUILD CRANKO LOWER
PCUT ELEVON LR
ACCESS ELEVON LR
MCUT
BUILD TIP LOWER
PCUT: ELEVON LT
ACCESS ELEVON LT
MCUT
BUILD VT1 LOWER
PCUT RUDDER LT
ACCESS RUDDER LT
MCUT

ORIGINAL PAGE IS OF POOR QUALITY

GTM INPUT--ELEVON AND RUDDER SURFACES
C. INSURF WING SEGMENTS HAVE 13 POINTS

C. ACCESS CRANKO UPPER
NSEG = 17
EQARC
ACCESS CRANKO LOWER
NSEG = 17
EQARC
ACCESS TIP UPPER
NSEG = 17
EQARC
ACCESS TIP LOWER
NSEG = 12
EQARC
ACCESS VT1 UPPER
NSEG = 17
EQARC
ACCESS VT1 LOWER
NSEG = 12
EQARC

C. REDEFINE ELEVON AND RUDDER SEGMENTS WITH 2 POINTS

C. ACCESS ELEVON UR
NSEG = 1
EQARC
ACCESS ELEVON LR
NSEG = 1
EQARC
ACCESS ELEVON UT
NSEG = 1
EQARC
ACCESS ELEVON LT
NSEG = 1
EQARC
ACCESS RUDDER UT
NSEG = 1
EQARC
ACCESS RUDDER LT
NSEG = 1
EQARC

EXIT

GTM INPUT--ELEVON AND RUDDER SURFACES (cont.)
SECTION EDIT

C. BUILD ELEVON COMPONENT

C. ACCESS VEHICLE ORBITER
    BUILD = ACCESS
    BUILD COMPONENT ELEVON UPPER
    BUILD SECTION ROOT
    COPY SEGMENT ELEVON UT
    BUILD SECTION TIP
    COPY SEGMENT ELEVON UT
    BUILD COMPONENT ELEVON LOWER
    BUILD SECTION TIP
    COPY SEGMENT ELEVON LT
    BUILD SECTION ROOT
    COPY SEGMENT ELEVON LR

C. BUILD RUDDER COMPONENT

C. BUILD COMPONENT RUDDER UPPER
    BUILD SECTION ROOT
    COPY SEGMENT ELEVON UT
    BUILD SECTION TIP
    COPY SEGMENT RUDDER UT
    BUILD COMPONENT RUDDER LOWER
    BUILD SECTION TIP
    COPY SEGMENT RUDDER LT
    BUILD SECTION ROOT
    COPY SEGMENT ELEVON LT

C. CORRECT WING SECTIONS TO ACCEPT RUDDER AND ELEVON

C. ACCESS COMPONENT WING UPPER
    ACCESS SECTION CRANK
    REPLACE SECTION CRANKO UPPER
    ACCESS SECTION TIP
    REPLACE SECTION TIP UPPER
    ACCESS SECTION VT1
    INSERT SECTION VT1 UPPER
    ACCESS COMPONENT WING LOWER
    ACCESS SECTION CRANK
    INSERT SECTION VT1 LOWER
    ACCESS SECTION TIP
    REPLACE SECTION TIP LOWER
    ACCESS SECTION CRANK
    REPLACE SECTION CRANKO LOWER
    TREE LIST ORBITER
    OUTPUT ORBITER

EXIT
    SAVE DATA BASE
    STOP

*EOF

GTM INPUT--ELEVON AND RUDDER SURFACES (cont.) 97
WING WITH FLAP, ELEVON, AND RUDDER
Step 5: Rotations and Deflections

The preceding four steps created the desired geometry in a flat wing. This step accomplishes the rotations such that the vertical control surfaces have a dihedral of 60°, and the outboard wing section has a dihedral of 15°. The flap has a deflection of 15°, the elevon has a deflection of 10°, and the rudder has a deflection of 25°. The following command sequence for the GTM is used for these operations:
EXECUTE GTM
C. ROTATE WING INTO THE DESIRED SHAPE
C. SECTION EDIT
ACCESS VEHICLE ORRITER
C. Rudder Deflection Rotation
C. ACCESS COMPONENT RUDDER UPPER
RCEN = -2576.0,-132
PITCH = -25
ROT
ACCESS COMPONENT RUDDER LOWER
ROT
C. Elevon Deflection Rotation
C. PITCH = 10
ACCESS COMPONENT ELEVON UPPER
ROT
ACCESS COMPONENT ELEVON LOWER
ROT
C. Flap Deflection Rotation
C. RCEN = -7476.0,-132
PITCH = 15
ACCESS COMPONENT FLAP UPPER
ROT
ACCESS COMPONENT FLAP LOWER
ROT
C. First Wing Rotation: V-T Portion Up 45 Degrees
C. RCEN = -2676.954,-132
PITCH = 0
ROLL = 45
ROLL. = -45
ACCESS COMPONENT RUDDER UPPER
ROT
ACCESS COMPONENT RUDDER LOWER
ROT
ACCESS COMPONENT WING UPPER
START TIP UPPER
ROT
ACCESS COMPONENT WING LOWER
STOP TIP LOWER
ROT

ORIGINAL PAGE IS
OF POOR QUALITY
SECOND WING ROTATION--ALL OUTBOARD OF CRANK UP 15 DEGREES:

\[
\begin{align*}
\text{RCEN} & = -2676.573 + 132 \\
\text{ROLL} & = 15 \\
\text{ROLL} & = -15 \\
\text{ACCESS COMPONENT Rudder Upper Rot.} & \\
\text{ACCESS COMPONENT Rudder Lower Rot.} & \\
\text{ACCESS COMPONENT Elevon Upper Rot.} & \\
\text{ACCESS COMPONENT Elevon Lower Rot.} & \\
\text{ACCESS COMPONENT Wing Upper Start = Cranko Upper Rot.} & \\
\text{ACCESS COMPONENT Wing Lower Stop = Cranko Lower Rot.} & \\
\text{Tree List Orbiter Output Orbiter} & \\
\text{EXIT} & \\
\text{SAVE DATA BASE} & \\
\text{STOP} & \\
\text{*EOF} & \\
\end{align*}
\]

GTM INPUT--ROTATIONS AND DEFLECTIONS (cont.)
GULL WING
GULL WING
BODY ENHANCEMENT

A minimum envelope body composed of 6 sections was generated using a panel program. The cross-sections represent the minimum area needed to accommodate the internal tank arrangements. The nose portion of this body was enhanced by generating new sections using the power law distribution with an exponent of 0.6. The second section of the original body was selected as the base section for this distribution. The newly generated sections are scaled versions of the base section at the following locations:

<table>
<thead>
<tr>
<th>Station-x</th>
<th>Magnification Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0.0166</td>
</tr>
<tr>
<td>-5</td>
<td>0.0435</td>
</tr>
<tr>
<td>-10</td>
<td>0.0659</td>
</tr>
<tr>
<td>-20</td>
<td>0.0999</td>
</tr>
<tr>
<td>-50</td>
<td>0.1731</td>
</tr>
<tr>
<td>-100</td>
<td>0.2624</td>
</tr>
<tr>
<td>-150</td>
<td>0.3333</td>
</tr>
<tr>
<td>-200</td>
<td>0.3977</td>
</tr>
<tr>
<td>-500</td>
<td>0.6891</td>
</tr>
</tbody>
</table>

The command sequence for the GTM used for this enhancement is as follows:
EXECUTE GMT
IMAGE INPT
GEOMETRY GENERATION
DATA BACE AND BODY
V=ARC BODV
C= CRY
C=0
C=
GEC
EXIT
EXIT

VEHICLE EDIT
FACE LIST AND BODY
ACCESS VEHICLE A C BODY
ACCESS EQUIPMENT BODY
BUILD SEGMENT BASE
COPY SECTION SECTION
EXIT

SEGMENT EDIT
L.BASE
XRC = 1
XRC =
SC BASE
BUILD SEGMENT NF
XRC = 1.18E
SC BASE
XMOVE = -1
MOVE
BUILD SEGMENT NF
XRC = 1.18E
SC BASE
XMOVE = -1
MOVE
BUILD SEGMENT NF
XRC = 1.18E
SC BASE
XMOVE = -1
MOVE
BUILD SEGMENT NF
XRC = 1.000
SC BASE
XMOVE = -2
MOVE
BUILD SEGMENT NF
XRC = 1.791
SC BASE
XMOVE = -2
MOVE

GMT INPT -- BODY ENHANCEMENT

ORIGINAL PAGE IS
OF POOR QUALITY
BUILD SEGMENT MFG 6
MAG = .2574
SC BASE
XMOVE = -100
MOVE
BUILD SEGMENT MFG 7
MAG = .2677
SC BASE
XMOVE = -200
MOVE
BUILD SEGMENT MFG 8
MAG = .6991
SC BASE
XMOVE = -500
MOVE
BUILD SEGMENT MFG 9
MAG = .723
SC BASE
XMOVE = -150
MOVE
EXIT

VEHICLE EDIT
ACCESS VEHICLE ARC BODY
ACCESS COMPONENT ARC BODY
ACCESS SECTION SECTION 1
INSERT SECTION NEW 1
ACCESS SECTION SECTION 1
INSERT SECTION NEW 2
ACCESS SECTION SECTION 1
INSERT SECTION NEW 3
ACCESS SECTION SECTION 1
INSERT SECTION NEW 4
ACCESS SECTION SECTION 1
INSERT SECTION NEW 5
ACCESS SECTION SECTION 1
INSERT SECTION NEW 6
ACCESS SECTION SECTION 1
INSERT SECTION NEW 7
ACCESS SECTION SECTION 2
INSERT SECTION NEW 8
ACCESS SECTION SECTION 1
REPLACE SECTION NEW 9
THEE LIST ARC BODY
OUTPUT ARC BODY
EXIT

SAVE DATA BASE
STOP
REOF

GTM INPUT -- BODY ENHANCEMENT (cont.)
ORIGINAL MINIMUM ENVELOPE BODY

ENHANCED MINIMUM ENVELOPE BODY