NASA CR-
141594

AEROPHYSICS RESEARCH CORPORATION
TECHNICAL NOTE

JTN-09
VOLUME I

(NASA-CR-141594) GEOMETRY TECHNOLOGY MODULE (GTM). VOLUME 1: ENGINEERING DESCRIPTION AND UTILIZATION MANUAL Final Report (Aerophysics Research Corp., Houston Tex.) 113 p HC $5.25 CSCL 09B G3/61

N75-17120

Unclas 09641

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

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-		N75-17120

4. Title and Subtitle	5. Report Date
GEOMETRY TECHNOLOGY MODULE (GTM) VOLUME I - ENGINEERING DESCRIPTION AND UTILIZATION MANUAL	December 1974
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
S. J. Reiners, G. N. Hirsch, G. E. Alford and C. R. Glatt	JTN-09 Volume I
9. Performing Organization Name and Address	10. Work Unit No.
Aerophysics Research Corporation Houston, Texas 77058	11. Contract or Grant No. NAS9-13584
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administraiton Johnson Spacecraft Center Engineering Analysis Division	Ex4

15. Supplementary Notes

Final Report

16. 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.

PRICES SUBJECT TO CHANGE

17. Key Words (Suggested by Author(s))	18. Distribution Statement
EDIN, geometry, design, technology, data base	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	107	

*For sale by the National Technical Information Service, Springfield, Virginia 22151

AEROPHYSICS RESEARCH CORPORATION
TECHNICAL NOTE

JTN-09
VOLUME I

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 I - Geometry Technology Module Engineering Description and Utilization Manual.

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

Preceding page blank

TABLE OF CONTENTS (Continued)

GEOMETRY TECHNOLOGY MODULE (GTM)

VOLUME I - ENGINEERING DESCRIPTION AND UTILIZATION MANUAL

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

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.

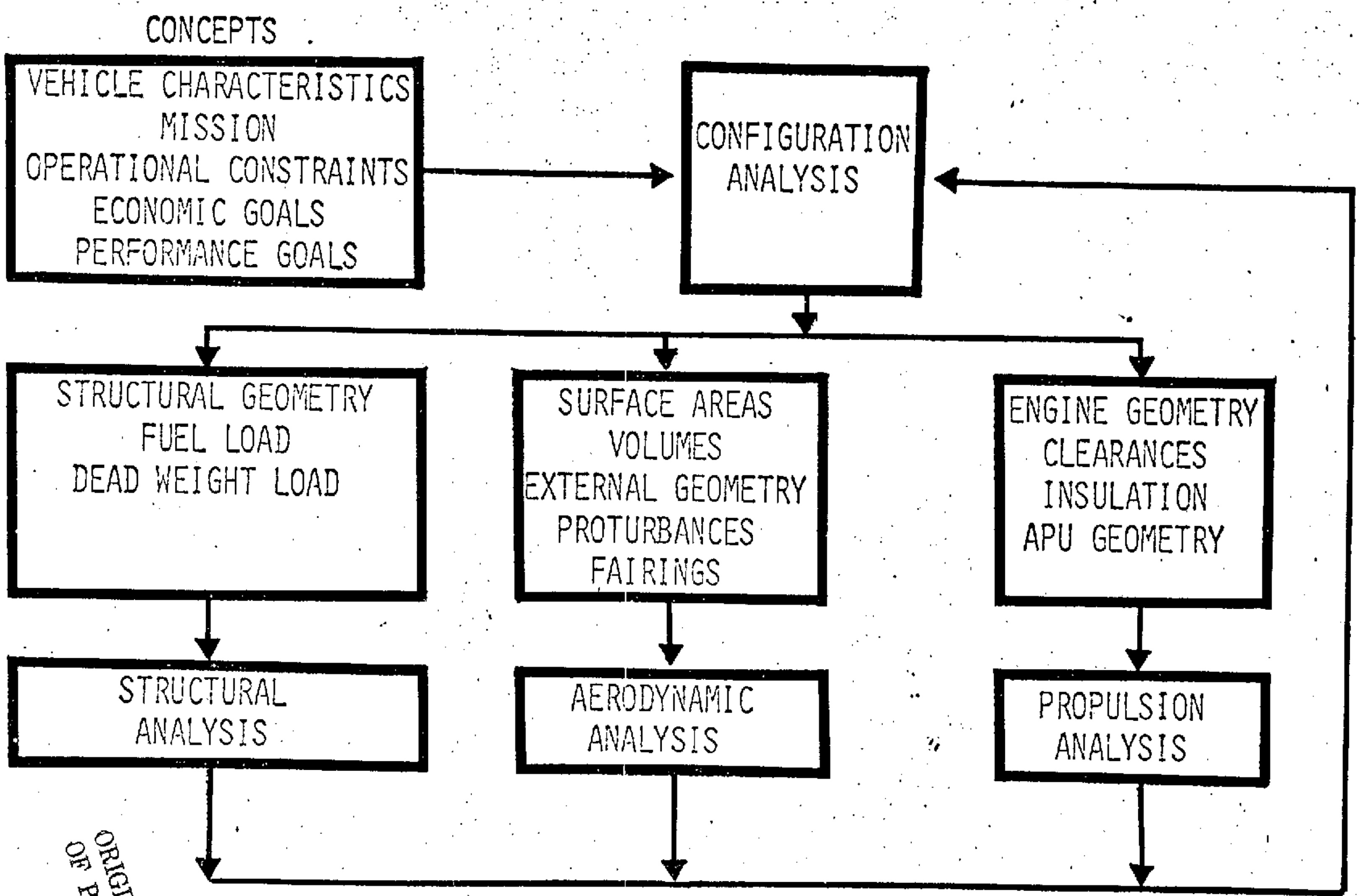


FIGURE 1 CONFIGURATION ANALYSIS

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.

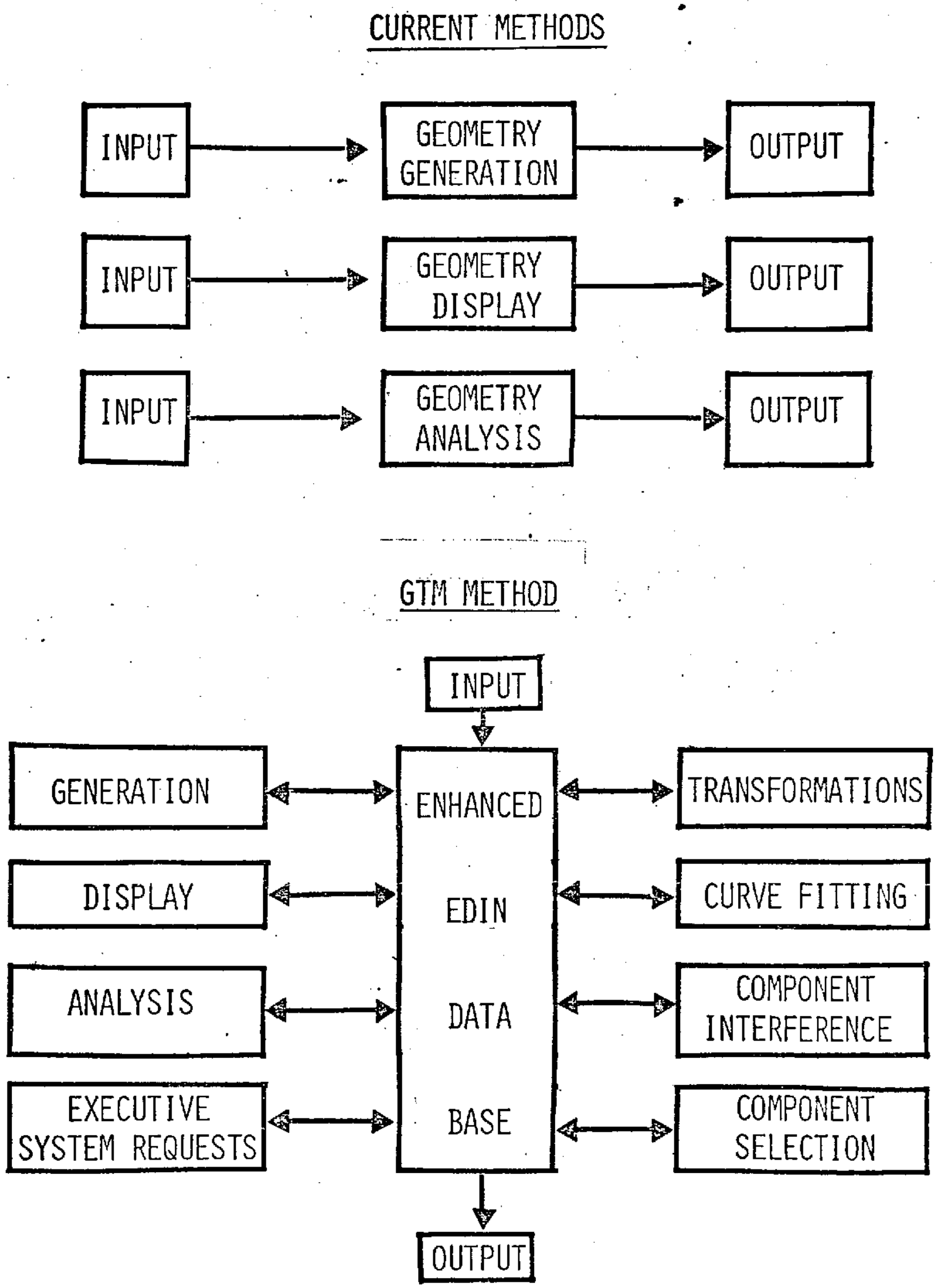


FIGURE 2 COMPARISON OF CONFIGURATION ANALYSIS METHODS

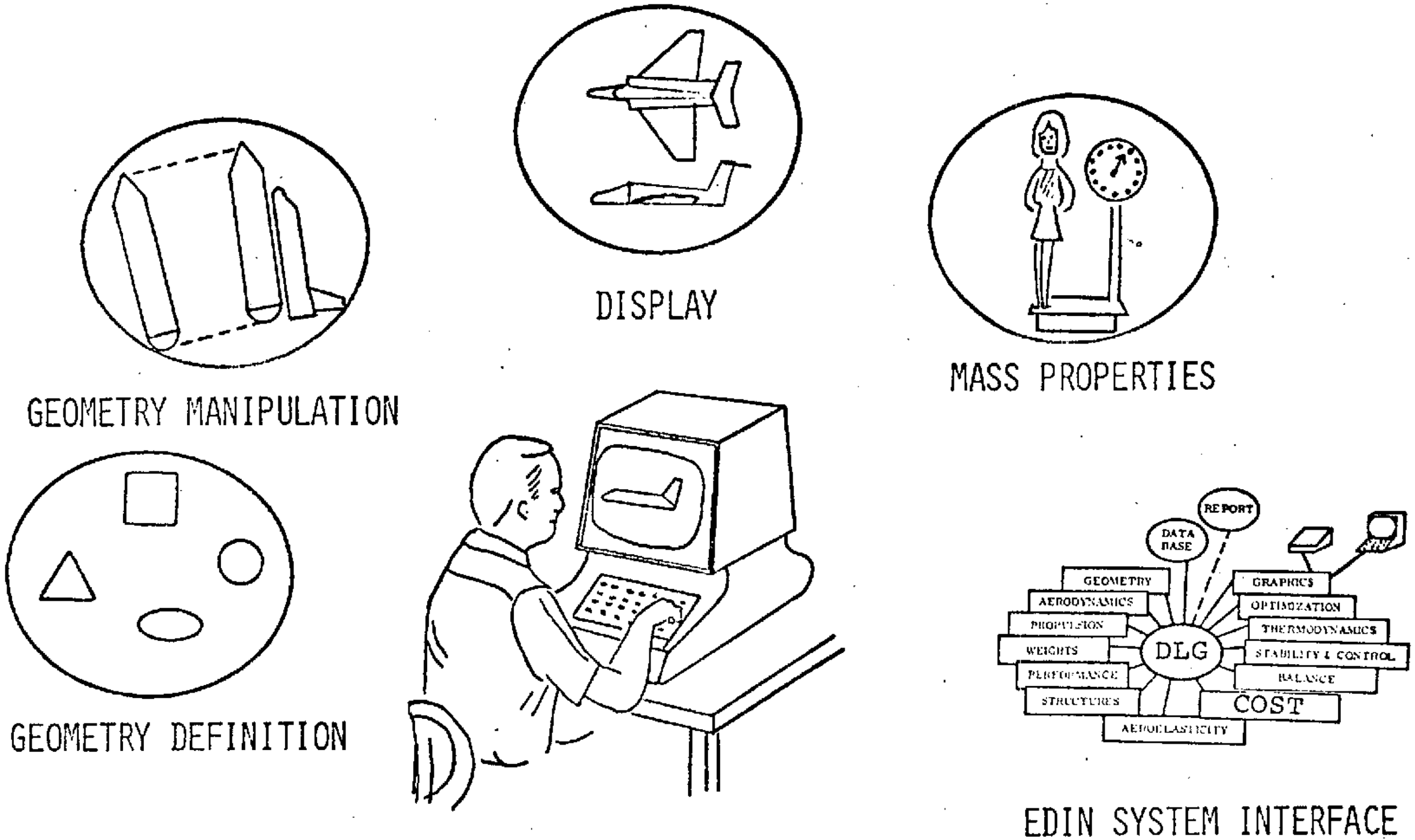


FIGURE 3 GTM CAPABILITIES.

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 viewers 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

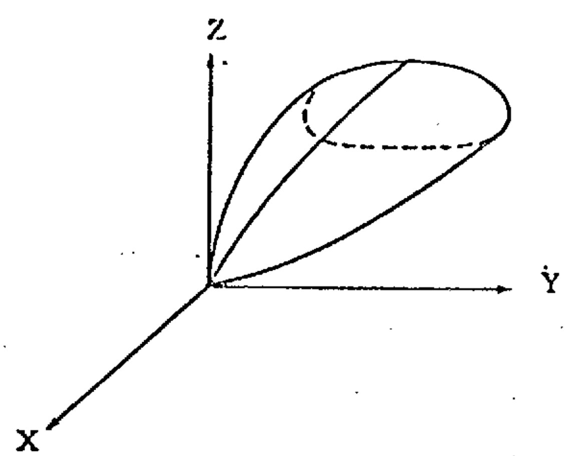
The display model comes to the stored geometry to quadrilateral elements then scales, translates and rotates the geometry elements to the desired position and orientation with respect to the viewer. The resulting plot vector is directed to the target display device.

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.

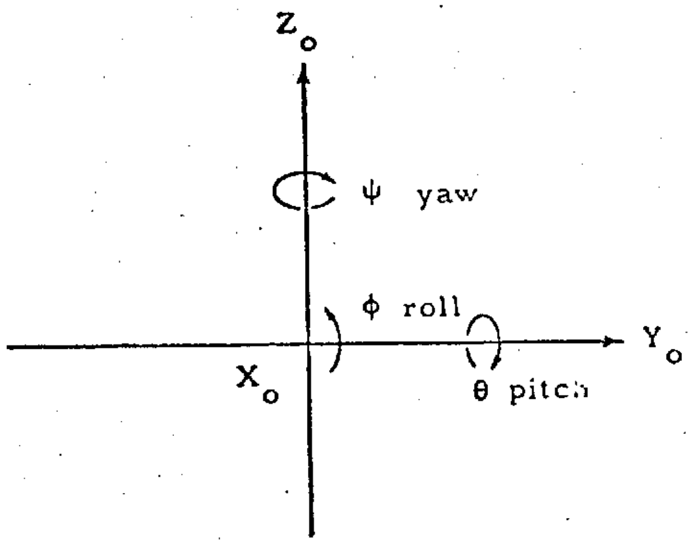
Coordinate System. - Each point on the surface is described by its coordinates in the body reference coordinate system.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{1}$$

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



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 (ψ, θ, ϕ). The rotation is given by the following relationship:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \phi \end{bmatrix} \begin{bmatrix} \theta \end{bmatrix} \begin{bmatrix} \psi \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad (2)$$

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 ϕ, θ and ψ are given by:

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

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

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

or

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

where

$$[E] = [\phi][\theta][\psi] \quad (7)$$

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.

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

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_o, Y_o, Z_o to the desired position with respect to the system reference axes. The complete transformation is shown below:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} E \end{bmatrix} \begin{bmatrix} x - X_o \\ y - Y_o \\ z - Z_o \end{bmatrix} \qquad (9)$$

<u>Mathematical Aids</u>. - 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_o direction (out of the plane of the paper) may be found from the following equation:

$$n_{x_o} = n_x(\cos\theta\cos\psi) + n_y(-\sin\psi\cos\phi + \sin\theta\cos\psi\sin\phi)$$

$$+ n_z(\sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi) \qquad (10)$$

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 of the angle α that the unit normal makes with the plane of the paper.

$$\alpha = \cos^{-1}(n_{x_o}) \qquad (11)$$

If the angle α is restricted as follows:

$$0 < \alpha < 5 \text{ (degrees)} \qquad (12)$$

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 α 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 compuers. 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.

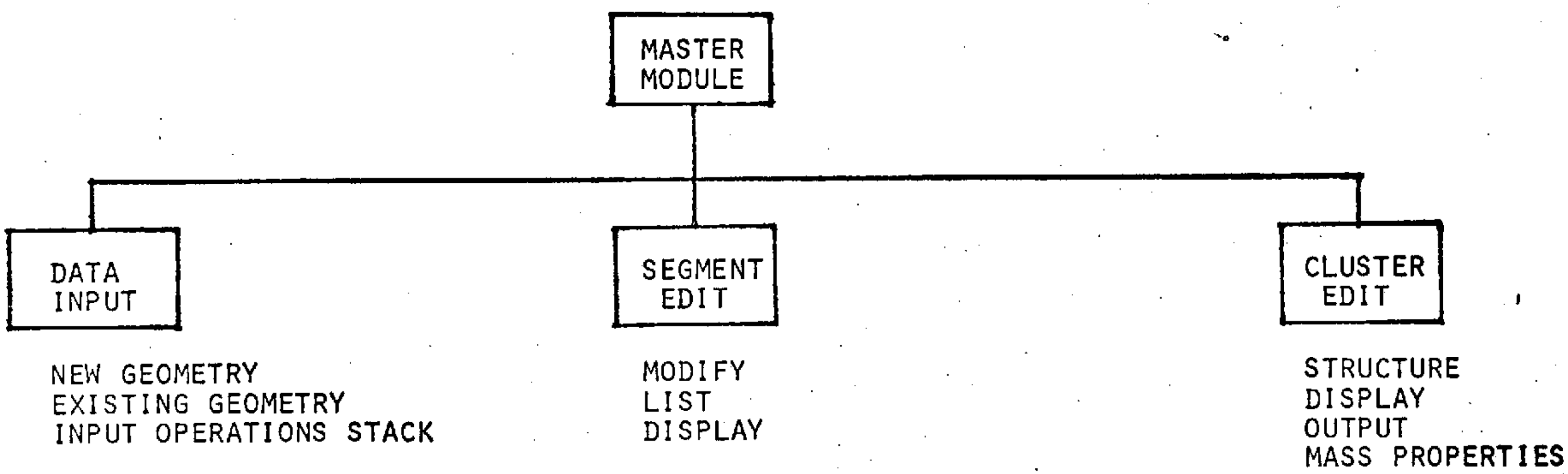


FIGURE 5 GTM PROGRAM ORGANIZATION.

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 data base 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 data base 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.

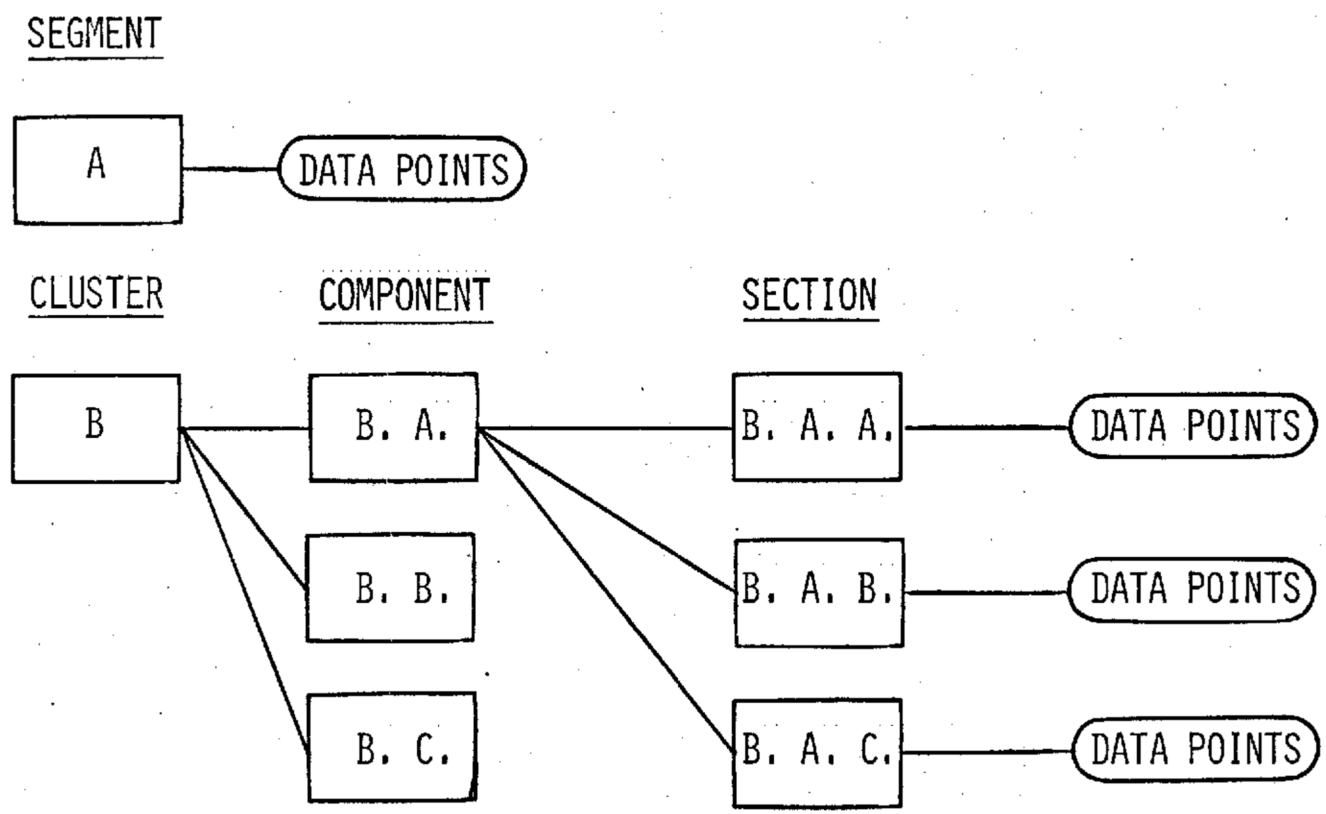


FIGURE 6 GTM GEOMETRY STRUCTURES.

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.

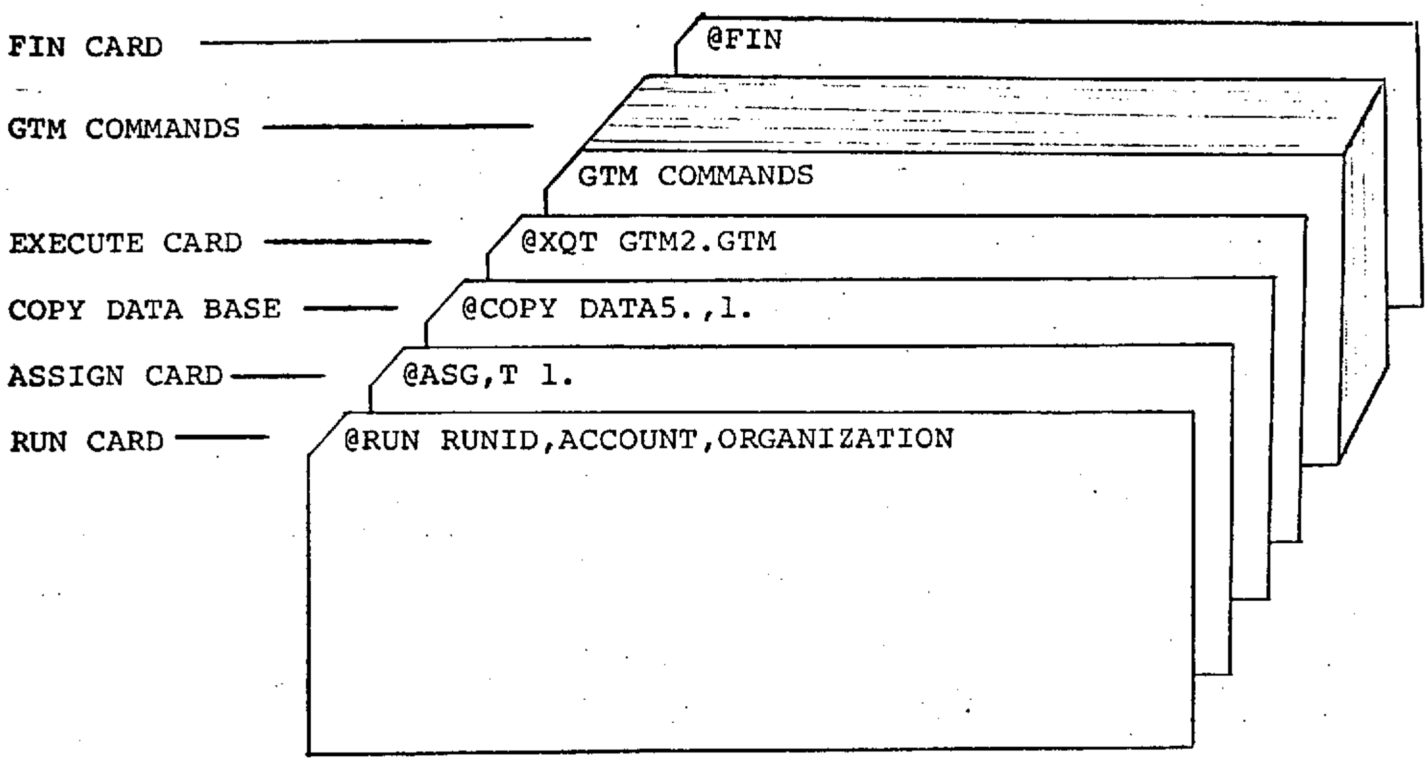


FIGURE 7 TYPICAL RUN STREAM.

Program Input

The GTM operates upon a data base 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 data base 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 / VEHICLE] = __:__:__

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

⋮

SECTION n

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.

<u>Edit Command Summary</u>. -

BUILD	[CLUSTER / COMPONENT / SECTION]	__:__:__
BUILD	[SEGMENT]	__:__:__
ACCESS	[CLUSTER / COMPONENT / SECTION / SEGMENT]	__:__:__
LOCATE	[CLUSTER / COMPONENT / SECTION]	__:__:__
INSERT	[CLUSTER / COMPONENT / SECTION]	(__:__:__)
DELETE	[CLUSTER / COMPONENT / SECTION]	(__:__:__)
REPLACE	[CLUSTER / COMPONENT / SECTION]	(__:__:__)
COPY	[CLUSTER / COMPONENT / SECTION]	(__:__:__)
COPY	[SEGMENT]	(__:__:__)
ADD	[CLUSTER / COMPONENT / SECTION]	(__:__:__)

Output Command Summary. -

```
[LIST]    [CLUSTER  ]
[ L  ]    [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]
```

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]

Miscellaneous Commands

MENU

OMIT

EXIT

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.

BUILD [CLUSTER / COMPONENT / 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 [CLUSTER / COMPONENT / SECTION / SEGMENT] __:__:__

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 [CLUSTER / COMPONENT / SECTION] __:__:__

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.

INSERT [CLUSTER / COMPONENT / SECTION] (__:__:__)

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 in front of the position specified by the access.

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.

DELETE [CLUSTER / COMPONENT / SECTION] (__:__:__)

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.

REPLACE [CLUSTER / COMPONENT / SECTION] (__:__:__)

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.

COPY [CLUSTER / COMPONENT / SECTION] (__:__:__)

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.

COPY SEGMENT (__:__:__)

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    (__:__:__)
```

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.

[RVEC ROTATION VECTOR] = __,__,__

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.

[ROT ROTATION COMMAND]

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 rotation parameters must have been established before this command is issued. The default parameters for all parameters are zero. The values established are saved and need not be changed for subsequent and identical rotations.

Scaling Parameters: - The parameter commands are:

MAG = ___

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

XMAG = ___

This command sets the X scale factor.

YMAG = ___

This command sets the Y scale factor.

ZMAG = ___

This command sets the Z scale factor.

[MAGC MAGNIFICATION CENTERS] = __,__,__

The scaling equations used are:

XOUT = (XIN - XC) *XMAG + XC

YOUT = (YIN - YC) *YMAG + YC

ZOUT = (ZIN - YC) *ZMAG + 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.

[SC
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.

[MOVE
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 [BUILD / ACCESS / LOCATE]

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

[BUILD / ACCESS / LOCATE] = [BUILD / ACCESS / LOCATE]

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:

*[P / POINT] = n

[F / FIND] (X,Y,Z)

[R / REPLACE] (X,Y,Z)

[I / INSERT] (X,Y,Z)

[D / DELETE] (X,Y,Z)

[DEF / DEFINE] (X,Y,Z)

[A / ADD] (X,Y,Z)

[P / POINT] = n: 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.

[F / FIND] (X,Y,Z): This command will locate the action position or point number of the point in the segment nearest the specified

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:

START X,Y,Z

STOP X,Y,Z

NSTART = n

NSTOP = n

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:

XMOVE = ___

YMOVE = ___

ZMOVE = ___

MOVE (__:__:__)

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:

XMAG = ___

YMAG = ___

ZMAG = ___

MAG = ___

[MAGC / MANIFICATION CENTER] = X_C, Y_C, Z_C

[SC / SCALE] (___,___,___)

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:

$$X_T = (X_I - X_C) * XMAG + X_C$$

$$Y_T = (Y_I - Y_C) * YMAG + Y_C$$

$$Z_T = (Y_I - Y_C) * ZMAG + Z_C$$

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 / YAW] = ___ (Degrees)

[THETA / PITCH] = ___ (Degrees)

[PHI / ROLL] = ___ (Degrees)

[RCEN / ROTATION CENTER] = X_C, Y_C, Z_C

[ROT / ROTATE] (___:___:___)

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.

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

XCUT = ___: This assumes a YZ Plane passing through the specified X with a direction of positive X.

YCUT = ___: This assumes a XZ Plane passing through the given Y value in the direction of positive Y.

ZCUT = ___: This assumes a XY Plane passing through the given Z with a direction of positive Z.

PCUT (__:__:__): This command will cause all of the points in the direction of the positive normal, plus all plane intersections to be output.

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:

NSEG = ___

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:

*ACCESS CLUSTER __:__:__

*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.

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

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

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

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

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

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

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

Segment Level Commands. - 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:

[L / 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:

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

How to Call a Menu

SAMPLE CASE 1: A menu of available commands may be obtained at any time by commanding the following:

COMMAND	DESCRIPTION
MENU	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 SO147B	Will access the cluster named SO147B.
ACCESS COMPONENT WING	Will access the component named Wing of Cluster SO147B.
ACCESS SECTION SECTION 2	Will access the second section of component Wing of Cluster SO147B.

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 . . NAMEn	User specifies geometry name, e. g., RED1, WHITE2, BLUE3.
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)

COMMAND	DESCRIPTION
IMAGE OUTPUT SO147B	Will output the SO147B geometry in Gentry format to File 3.

How to Rotate Geometry

SAMPLE CASE 5:

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

How to Translate Geometry

SAMPLE CASE 6:

COMMAND	DESCRIPTION
CLUSTER EDIT	Provides access to the cluster edit module.
XMOVE=10.	Specifies a X-translation of 10 inches.
YMOVE=-1500.	Specifies a Y-translation of -1500 inches.
ZMOVE=.001	Specifies a Z-translation of .001 inches.
ACCESS CLUSTER SO147B	Provides access to SO147B geometry.
ACCESS COMPONENT WING	Provides access to SO147B wing (only) geometry.
MOVE	Commands the move.

How to Display Geometry

SAMPLE CASE 7:

COMMAND	DESCRIPTION
DISPLAY SO147B	Specifies the display of SO147B geometry.

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 calculatiions.

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

1. Glatt, C. R., Hague, D. S., Watson, D. A.: ODINEX: An Executive Computer Program for Linking Independent Programs. Aerophysics Research Corporation. JTN-01. 1973.

2. Glatt, C. R.: IMAGE: A Computer Code for Generating Picture-Like Images of Aerospace Vehicles. Aerophysics Research Croporation. LTN-11. 1973.

3. Glatt, C. R. and Colquitt, W. N.: The DLG Processor - A Data Management Executive for the Engineering Design Integration (EDIN) System. Aerophysics Research Corporation. Technical Note JTN-10. 1974.

4. Glatt, C. R., Hirsch, G. N., Alford, G. E., Colquitt, W. N. and Reiners, S. J.: The Engineering Design Integration (EDIN) System. Aerophysics Research Corporation. Technical Note JTN-11. 1974.

APPENDIX A
GTM COMMAND AND FUNCTION SUMMARY

MASTER MODULE MENU INDEX

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.

Primary Command: STOP
Optional Command: NA
Description: A user entry of "STOP" will terminate all GTM activity.

Primary Command: SAVE DATA BASE
Optional Command: NA
Description: A user entry of "SAVE DATA BASE" will store and save all geometry files in the GTM data base for the current execution.

Primary Command: OPS STACK name:name:name:
Optional Command: NA
Description: A user entry of "OPS STACK name:name: name, will access the operations stack stored in the GTM data base under the specified name:name:name, and perform the commands or functions associated within it.

Primary Command: EXTERNAL cc=name:name:name OS=name:name: name
Optional Command: NA
Description: A user entry of "EXTERNAL cc= n : n : n OS= n : n : n " transfers control from the GTM to an external control card stack in the data base. When GTM is reentered, the control stack specified by OS= n : n : n will be executed.

Primary Command: MIMIC
Optional Command: NA
Description: This command is mainly used in conjunction with "canned" operations. A user entry of "MIMIC" will display the user input for each computer inquiry.

Primary Command: HUSH
Optional Command: NA
Description: A user entry of "HUSH" will negate the MIMIC command and disallow the display of user inputs to computer inquiries. This command is mainly used in conjunction with "canned" operations.

IMAGE INPUT MODULE LANGUAGE INDEX

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 formt.

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

Primary Command:	DATA BASE name:name:name
Optional Command:	NA
Description:	A user entry of "DATA BASE name:name:name" will load from the data base the Gentry file under the specified name:name:name.
Primary Command:	BINARY FILE UNIT
Optional Command:	NA
Description:	In response to a computer inquiry for the unit to be read, a user entry of "BINARY FILE UNIT" will specify the named binary file to be read.
Primary Command:	BCD FILE UNIT
Optional Command:	NA
Description:	In response to a computer inquiry for the unit to be read, a user entry of "BCD FILE UNIT" will specify the named BCD file to be read.
Primary Command:	MENU
Optional Command:	NA
Description:	A user entry of "MENU" will generate a listing of the available commands and functions of the Gentry input module.
Primary Command:	EXIT
Optional Command:	FINISHED INPUT COMPLETE INPUT COMPLETED
Description:	A user entry of "EXIT" or any of its options will cause an exit from the Gentry module to the master module.

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.

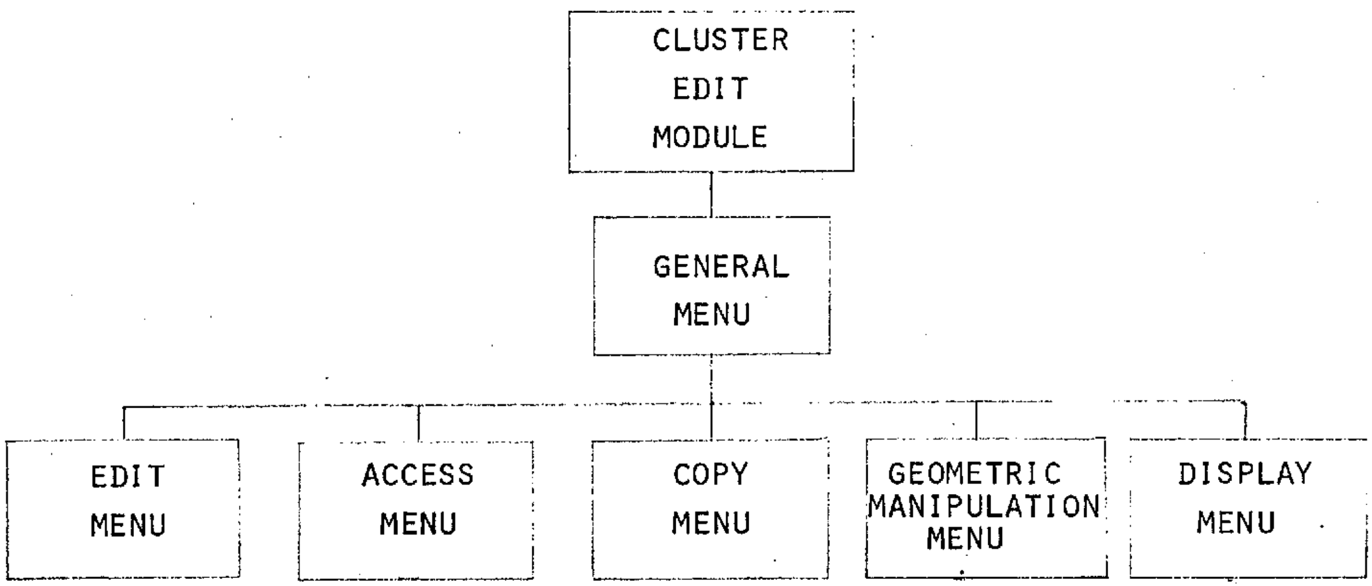
Primary Command: COMPONENT name: name: name
Optional Command: COMPONENT NAME name: name: name
COM name: name: name
Description: A user entry of "COMPONENT name: name: name" in response to a GTM inquiry for a component name will assign the input name: name: name to the geometry.

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.

CLUSTER EDIT MODULE MENU INDEX

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:



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.

Primary Command: MENU EDIT
Optional Command: NA
Description: A user entry of "MENU EDIT" will provide a listing of the EDIT MENU.

Primary Command: MENU ACCESS
Optional Command: NA
Description: A user entry of "MENU ACCESS" will provide a listing of the ACCESS MENU.

Primary Command: MENU COPY
Optional Command: NA
Description: A user entry of "MENU COPY" will provide a listing of the COPY MENU.

Primary Command: MENU GEOMETRIC MANIPULATION
Optional Command: NA
Description: A user entry of "MENU GEOMETRIC MANIPULATION" will generate a listing of the MANIPULATION MENU.

Primary Command: MENU DISPLAY
Optional Command: NA
Description: A user entry of "MENU DISPLAY" will generate a listing of the MENU DISPLAY.

Primary Command: EXIT
Optional Command: NA
Description: A user entry of "EXIT" will terminate cluster edit activity and return to the master module.

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

Primary Command: BUILD CLUSTER name: name: name
Optional Command: BCL name: name: name
Description: A user entry of "BUILD CLUSTER name: name: name" will create a cluster with the specified name.

Primary Command: BUILD COMPONENT name: name: name
Optional Command: B COML name: name: name
Description: A user entry of "BUILD COMPONENT name: name: name" will create a component with the specified name.

Primary Command:	BUILD SECTION name: name: name
Optional Command:	B SEC name: name: name
Description:	A user entry of "BUILD SECTION name: name: name" will create a section with the specified name.
Primary Command:	BUILD SEGMENT name: name: name
Optional Command:	B SEG name: name: name
Description:	A user entry of "BUILD SEGMENT name: name: name" will create a segment with the specified name.
Primary Command:	INSERT CLUSTER name: name: name
Optional Command:	I CL name: name: name
Description:	A user entry of "INSERT CLUSTER name: name: name" will insert the named cluster in the currently accessed cluster.
Primary Command:	INSERT SECTION name: name: name
Optional Command:	I SEC name: name: name
Description:	A user entry of "INSERT SECTION name: name: name" will insert the named section ahead of the currently accessed section.
Primary Command:	INSERT SEGMENT name: name: name
Optional Command:	I SEG name: name: name
Description:	A user entry of "INSERT SEGMENT name: name: name" will insert the named segment ahead of the currently accessed segment.
Primary Command:	INSERT COMPONENT name: name: name
Optional Command:	I COM name: name: name
Description:	A user entry of "INSERT COMPONENT name: name: name" will insert the named component ahead of the currently accessed component.
Primary Command:	REPLACE SEGMENT name: name: name
Optional Command:	R SEG name: name: name
Description:	A user entry of "REPLACE SEGMENT name: name: name" will replace the currently accessed segment with the specified name.

Primary Command: REPLACE COMPONENT _name_: _name_: _name_
Optional Command: R COM _name_: _name_: _name_
Description: A user entry of "REPLACE COMPONENT _name_: _name_: _name_" will replace the currently accessed component with the specified name.

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

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

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

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

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

Primary Command: DELETE CLUSTER _name_: _name_: _name_
Optional Command: D CL _name_: _name_: _name_
Description: A user entry of "DELETE CLUSTER _name_: _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.

Primary Command: ACCESS SEGMENT name: name: name
Optional Command: AC SEG name: name: name
Description: A user entry of "ACCESS SEGMENT name: name: name" will access the named segment geometry from the data base.

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

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

Primary Command: LOCATE=ACCESS
Optional Command: LOC=A
Description: A user entry of "LOCATE=ACCESS" will set the locate register equal to the access register.

Primary Command: ZERO LOCATE
Optional Command: ZERO LOC
Description: A user entry of "ZERO LOCATE" will clear the locate register.

Primary Command: ZERO ACCESS
Optional Command: ZERO AC
Description: A user entry of "ZERO ACCESS" will clear the access register.

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

Primary Command: START X : Y : Z
Optional Command: NA
Description: A user entry of "START X : Y : Z " specifies a start point at the named X,Y,Z position.

Primary Command: STOP X : Y : Z
Optional Command: NA
Description: A user entry of "STOP X : Y : Z " specifies a stop point at the named X,Y,Z position.

Primary Command: COPY CLUSTER name: name: name
Optional Command: C CL name: name: name
Description: A user entry of "COPY CLUSTER name: name: name" will copy the component and sections of the accessed CLUSTER to the specified name.

Primary Command: COPY COMPONENT name: name: name
Optional Command: C COM name: name: name
Description: A user entry of "COPY COMPONENT name : name: name" will copy the sections of the accessed component to the specified component name.

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 segments of the accessed section to the named section.

Primary Command: COPY SEGMENT name: name: name
Optional Command: C SEG name: name: name
Description: A user entry of "COPY SEGMENT name: name: name" will copy the accessed segments to the named segment.

Primary Command: ADD SECTION name: name: name
Optional Command: A SEC name: name: name
Description: A user entry of "ADD SECTION name: name: name" will copy the accessed data pointer to the named component.

Primary Command: ADD SEGMENT name: name: name
Optional Command: A SEG name: name: name
Description: A user entry of "ADD SEGMENT name: name: name" will copy the accessed data pointer to the named section.

Primary Command: ADD COMPONENT name: name: name
Optional Command: A COM name: name: name
Description: A user entry of "ADD COMPONENT name: name: name" will copy the accessed data pointer to the named cluster.

Primary Command: ADD CLUSTER name: name: name
Optional Command: A CL name: name: name
Description: A user entry of "ADD CLUSTER name: name: name" will copy all accessed component pointers to the named cluster.

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 : Z : 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.

Primary Command: XMOVE=value
Optional Command: NA
Description: A user entry of "XMOVE=value" will translate the geometry X station values by the amount specified once the move command is given.

Primary Command: YMOVE=value
Optional Command: NA
Description: A user entry of "YMOVE=value" will translate the geometry Y station values by the amount specified once the move command is given.

Primary Command: ZMOVE=value
Optional Command: NA
Description: A user entry of "ZMOVE=value" will translate the geometry Z station values by the amount specified once the move command is given.

Primary Command: MOVE
Optional Command: NA
Description: A user entry of "MOVE" will command the program to execute the moves previously specified by XMOVE, YMOVE or ZMOVE.

Primary Command: XMAG=value
Optional Command: NA
Description: A user entry of "XMAG=value" specifies a X station magnification factor. This entry is used in conjunction with the scale command.

Primary Command: YMAG=value
Optional Command: NA
Description: A user entry of "YMAG=value" specifies a Y station magnification factor. This entry is used in conjunction with the scale command.

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

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
V 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).

SEGMENT EDIT MODULE MENU INDEX

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: name
Optional Command:	COPY name: name: name
Description:	A user entry of "COPY SEGMENT name: name: name" will copy the accessed segment geometry to the specified name.
Primary Command:	BUILD SEGMENT name: name: name
Optional Command:	BUILD name: name: name
Description:	A user entry of "BUILD SEGMENT name: name: name" will create a segment by the name specified.
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.
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.
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.
Primary Command:	ACCESS name: name: name
Optional Command:	AC name: name: name
Description:	A user entry of "ACCESS name: 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 a 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.

CALCULATOR MODULE MENU INDEX

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 ENT=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:	/ DIVIDE
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 Commdn:	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.

INPUT MODULE MENU INDEX

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:

Primary Command:	BCD INPUT name: name: name
Optional Command:	NA
Description:	A user entry of "BCD INPUT name: name: name" will store the card image data in the data base under the specified name.
Primary Command:	Numeric Input name: name: name
Optional Command:	NA
Description:	A user entry of "NUMERIC INPUT name: name: name" will enter numeric data under the specified name. The data is read in free field format.

APPENDIX B
GTM DATA BASE

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

NAME	DESCRIPTION
SO147B	SHUTTLE ORBITER CLUSTER
NOSE	NOSE CLUSTER
TAIL	TAIL CLUSTER
OMSSUB	OMS SUBTRACTION SURFACE
OMSPOD	OMS POD
OWING	OUTTER WING
IWING	INNER WING
BODY	BODY STRUCTURE
LE	WING LEADING EDGE
BFLAP	BODY FLAP
SRM	SOLID ROCKET BOOSTER
SHRDL	LOWER PAYLOAD SHROUD (LLV)
SHRDV	UPPER PAYLOAD SHROUD (LLV)
STAGE6	SIXTH STAGE LLV
STAGE5	FIFTH STAGE LLV
STAGE4	FOURTH STAGE LLV
STAGE3	THIRD STAGE LLV
STAGE2	SECOND STAGE LLV
STAGE1	FIRST STAGE LLV
LLV	LARGE LEFT VEHICLE

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

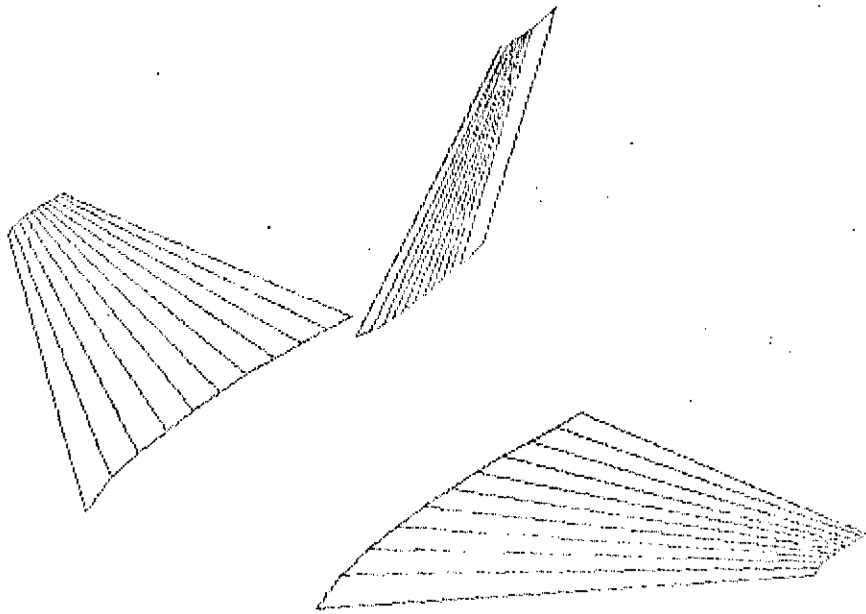
The standard airfoil geometry was generated by the AIRFOIL Program of reference 2 and passed to the GTM on a specially formatted file. The graphics representation was generated by the IMAGE Program of reference 2.

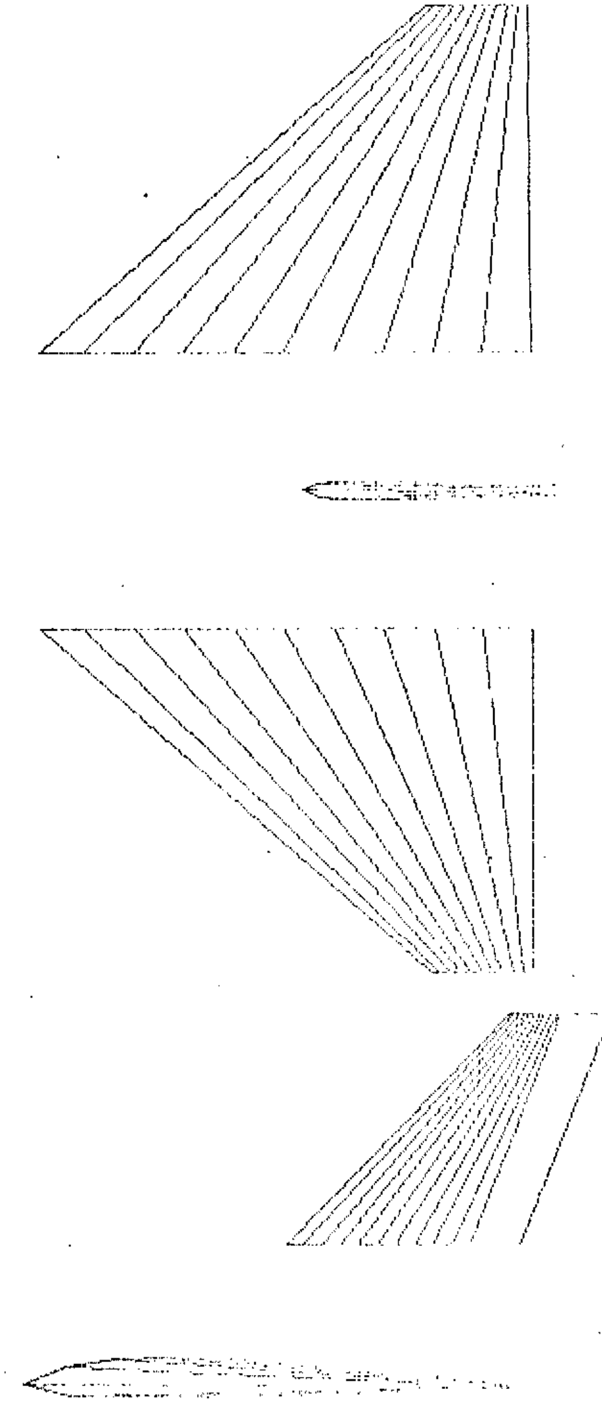
```
INPUT
        READ FILE 4
        EXIT
 SEGMENT EDIT
      L NACA 2408 UPPER
        NSEG=10
        EQARC NACA 2408 UPPER
        NSEG=10
        EQARC NACA 2408 LOWER
        NSEG=10
        EQARC NACA 2412 UPPER
        NSEG=10
        EQARC NACA 2412 LOWER
        MAG=9.72
        SC NACA 2408 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
      L NACA 2408 UPPER
        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 TIP
        COPY SEGMENT NACA 2412 LOWER
        BUILD SECTION ROOT
        COPY SEGMENT NACA 2408 LOWER
        TREE LIST PARALLEL BURN SSTO
        EXIT
```

WING GENERATION INPUT.

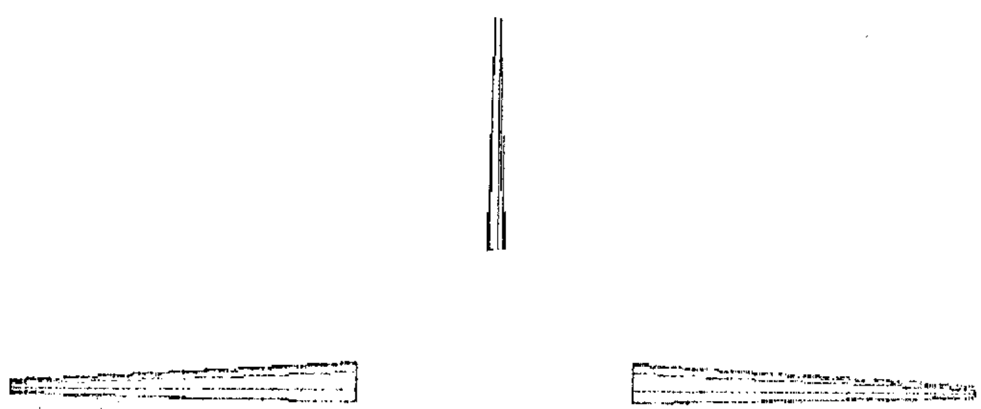
```
SEGMENT EDIT
        ACCESS NACA  0008 UPPER
        BUILD SEGMENT TIP
        COPY SEGMENT
        ACCESS  NACA  0008 UPPER
        BUILD SEGMENT ROOT
        COPY SEGMENT
        ACCESS ROOT
        NSEG = 10
        EQARC
        YAW = 180
        ROLL = -90
        ROLL =  90
        ROT
        MAG = 4.61
        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 SSTO
        BUILD = ACCESS
        BUILD COMPONENT VTAIL
        ADD SEGMENT TIP
        ADD SEGMENT ROOT
        TREE LIST PARALLEL BURN SSTO
        ACCESS VEHICLE PARALLEL BURN SSTO
        OUTPUT
        EXIT
 SAVE DATA BASE
 STOP
```

VERTICAL TANK GENERATION INPUT.





AERODYNAMIC SURFACE OUTPUT.



APPENDIX D
SAMPLE PROBLEM.

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: Createa 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:

Root	NACA 2408
Crank	NACA 2408
Tip	NACA 2412

These section 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. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.  BUILD FLAT CRANK WING
C. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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

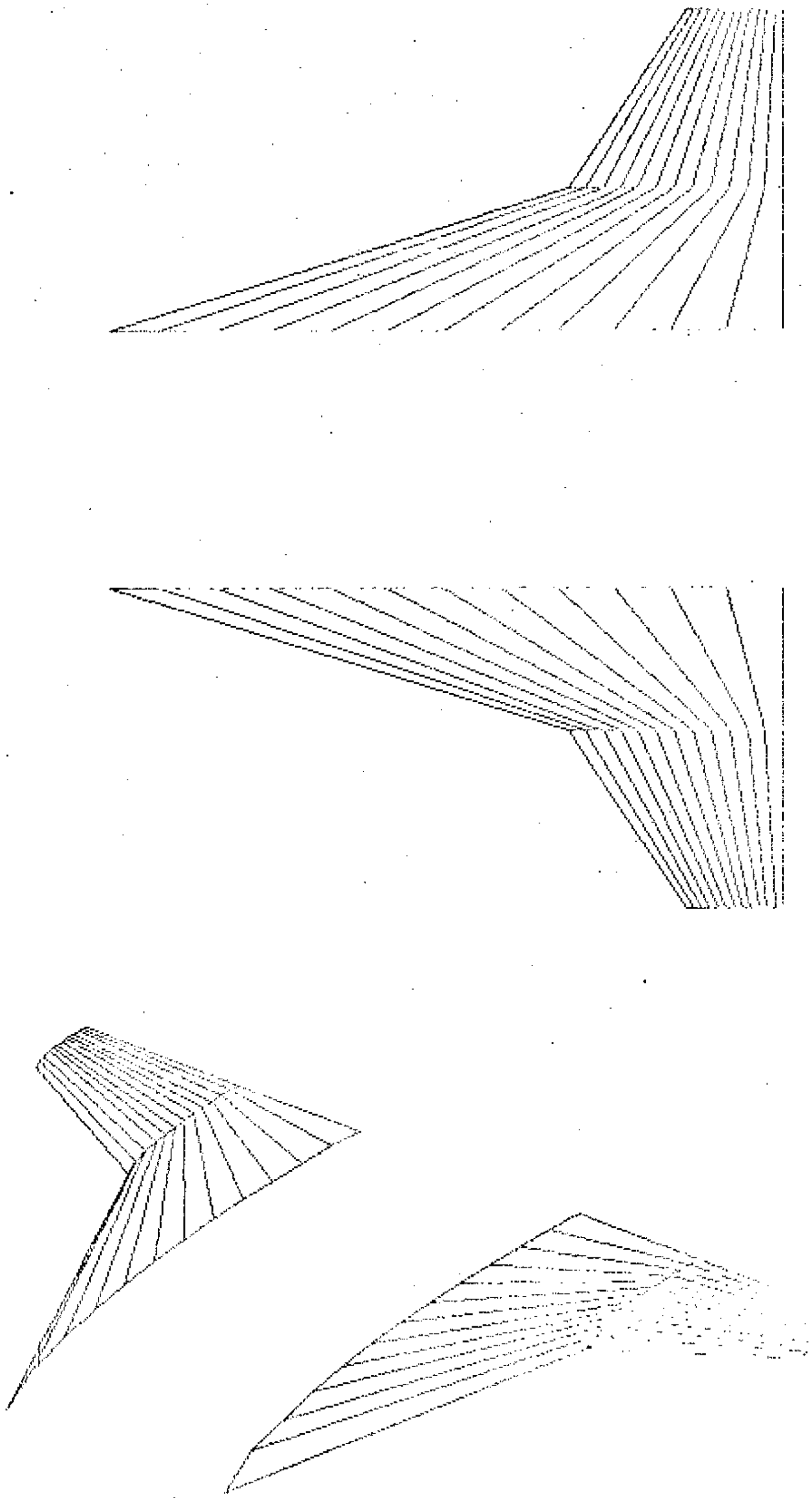
GTM INPUT--FLAT, CRANKED WING

```
C.
C.  SCALE A MOVE ROOT SEGMENTS
C.
      MAG=14.206
     XMOVE = -1256
     YMOVE = 272
     ZMOVE = -132
     SC ROOTU
     MOVE
     SC ROOTL
     MOVE
C.
C.  SCALE AND MOVE CRANK SEGMENTS
C.
     MAG = 4.51
      XMOVE = -2225
     YMOVE = 573
     SC CRANKU
     MOVE
     SC CRANKL
     MOVE
C.
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.
C.  SEGMENT MANIPULATIONS COMPLETE
C.
EXIT
```


GTM INPUT--FLAT, CRANKED WING (cont.)

```
SECTION EDIT
C.- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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.)



FLAT, CRANKED WING

Step 2: Vertical Stabilizer Surface

The vertical stabilzer surface is generated from the following sections:

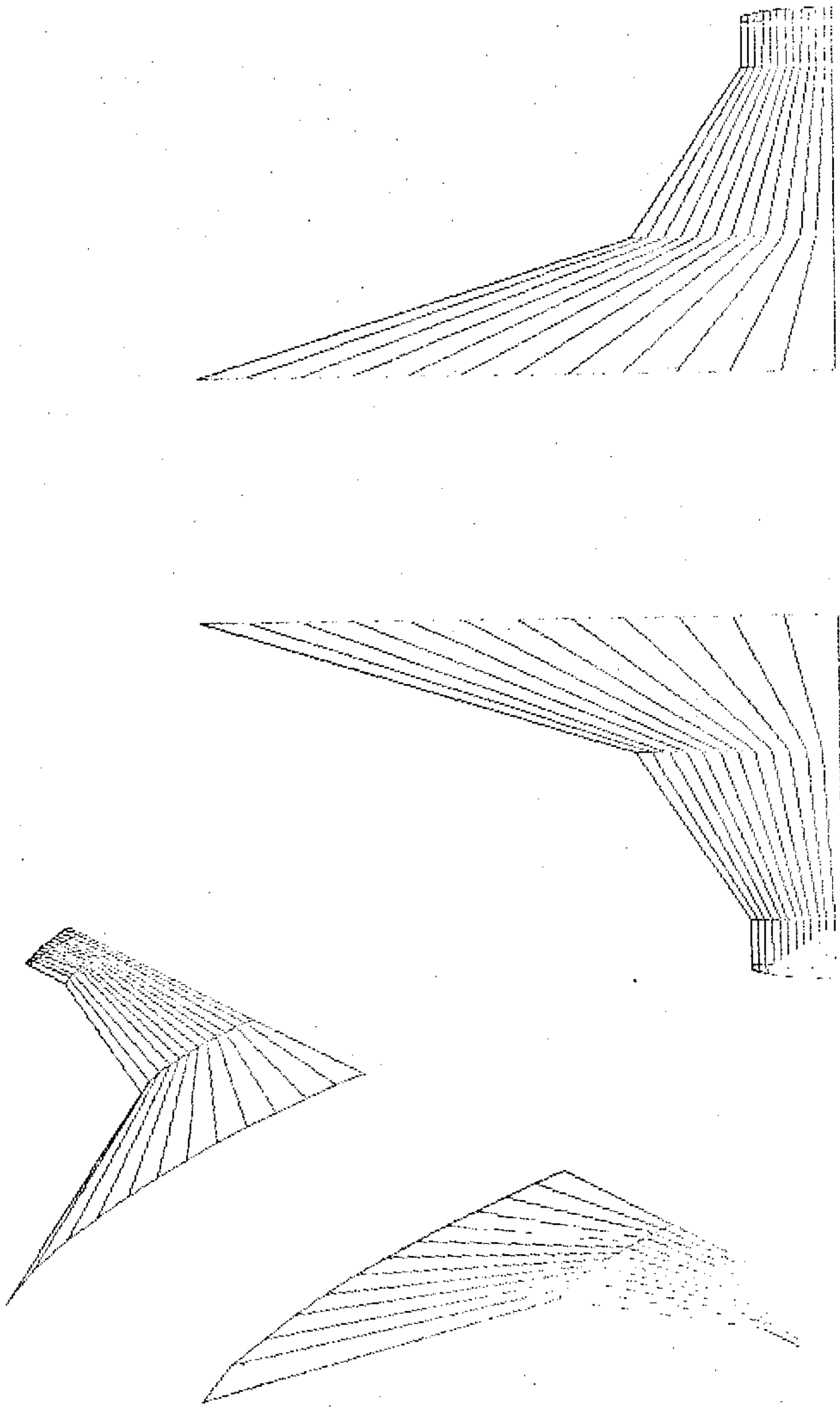
Tip	NACA 0008
Cap	Input section data

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≠
C.
C.- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.  BUILD WING TIP/VTAIL SECTION--FLAT
C.- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.
SEGMENT EDIT
C.
C.  ROTATE AND SCALE TIP SEGMENTS
C.
     ACCESS NACA 0008 UPPER
     NSEG = 12
     EQARC
      YAW = 180
     MAG = 2.01
     ROT
     SC
     ACCESS NACA 0008 LOWER
     NSEG = 12
     EQARC
     ROT
     SC
C.
C.  MOVE TIP SEGMENTS INTO DESIRED POSITION
C.
     XMOVE = -2475
      YMOVE = 1054
     ZMOVE = -132
     MOVE NACA 0008 UPPER
     MOVE NACA 0008  LOWER
C.
C.  BUILD TIP CAP SEGMENT
C.
      BUILD SEGMENT LINE
     ADD 0,0,0
     ADD -.2,.5,0
     ADD -.4,.8,0
     ADD -.6,.9,0
     ADD -1.,1.,0
C.
C.  SCALE AND MOVE CAP SEGMENT
C.
     XMAG = 201
     YMAG = 20.
     ZMAG = 1.
     SC
      XMOVE = -2475
     YMOVE = 1064
     ZMOVE = -132
     MOVE
     NSEG = 12
     EQARC
```

```
C.
C.  CREATE UNIQUE NAMES FOR SEGMENTS TO BE INSERTED
C.
      BUILD SEGMENT VT1
     COPY NACA 0008 LOWER
      BUILD SEGMENT VT2
     COPY LINE
EXIT
SECTION EDIT
C.- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.  ADD TAIL SECTIONS TO EXISTING WING
C.- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     ACCESS VEHICLE ORBITER
     ACCESS COMPONENT WING UPPER
C.
C. SET UP THE BUILD REGISTER
C.
     BUILD = ACCESS
     BUILD SECTION VT1
      COPY SEGMENT NACA 0008 UPPER
      BUILD SECTION VT2
     COPY SEGMENT LINE
     ACCESS COMPONENT WING LOWER
C.
C.  LOWER TIP SECTIONS MUST BE INSERTED TO BE PLACED IN THE
C.  POSITON.
C.
      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

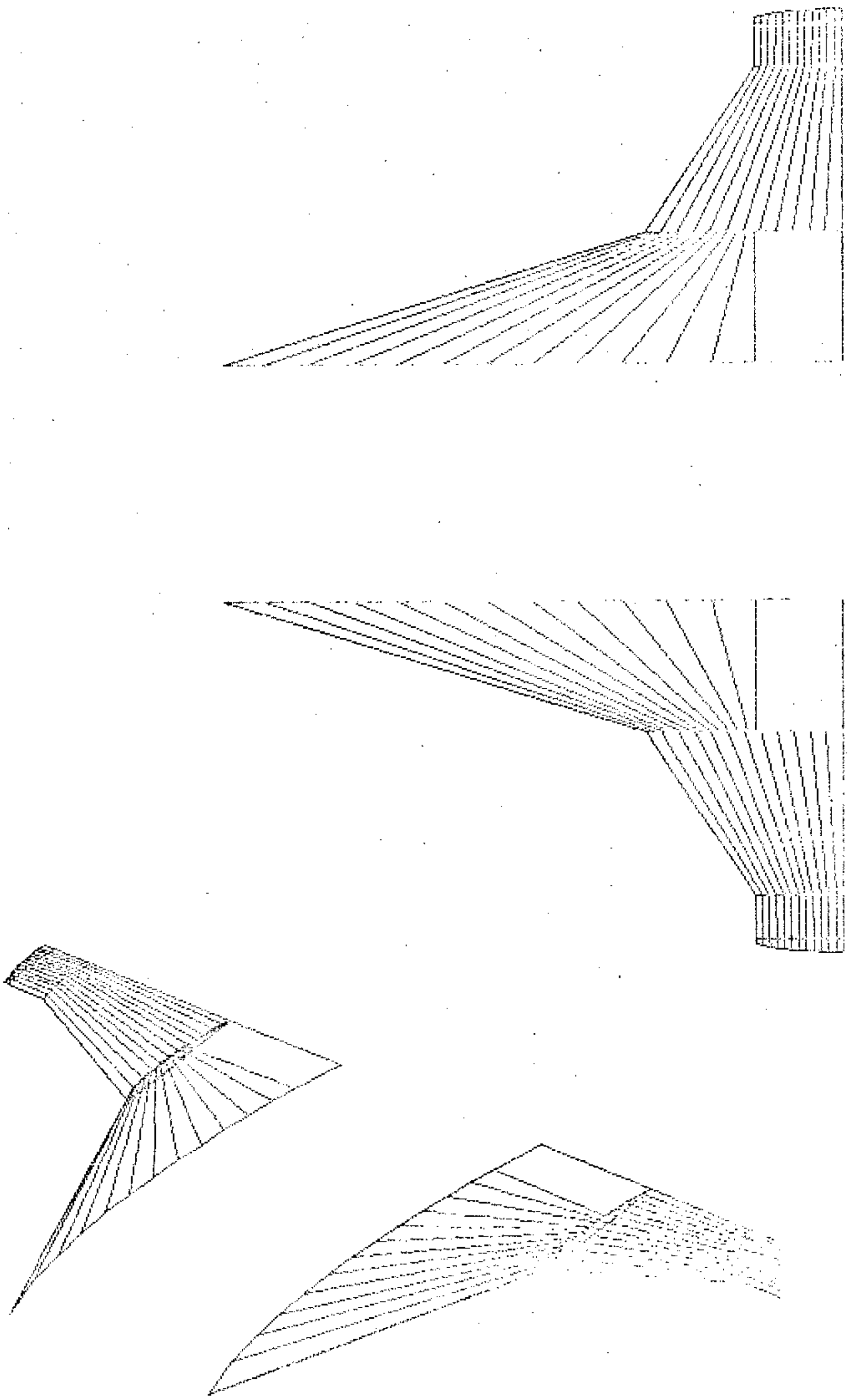
Step 3: Flap Surface

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.
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.
C.  FLAP SEGMENTS WILL BE BUILT FROM THE NEGATIVE PORTIONS OF THE
C.  SEGMENTS AS REFERENCED FROM THE CUTTING PLANE.
C.
C.  NEW WING SEGMENTS WILL BE THE POSATIVE 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.
C.  INSURE 12 POINTS FOR WING SEGMENTS
C.
     ACCESS ROOT UPPER
     NSEG = 12
     EQARC
     ACCESS ROOT LOWER
     NSEG = 12
     EQARC
```

```
      ACCESS CRANKI UPPER
      NSEG = 12
      EQARC
      ACCESS CRANKI LOWER
      NSEG = 12
      EQARC
C.
C.  EXTABLISH 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.  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 FLAPUT
      BUILD COMPONENT FLAP LOWER
      BUILD SECTION TIP
      COPY SEGMENT FLAPLT
      BUILD SECTION ROOT
      COPY SEGMENT FLAPLR
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.)



WING WITH FLAP

Step 4: Elevon and Rudder Surfaces

The Elevon is created by cuting 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:

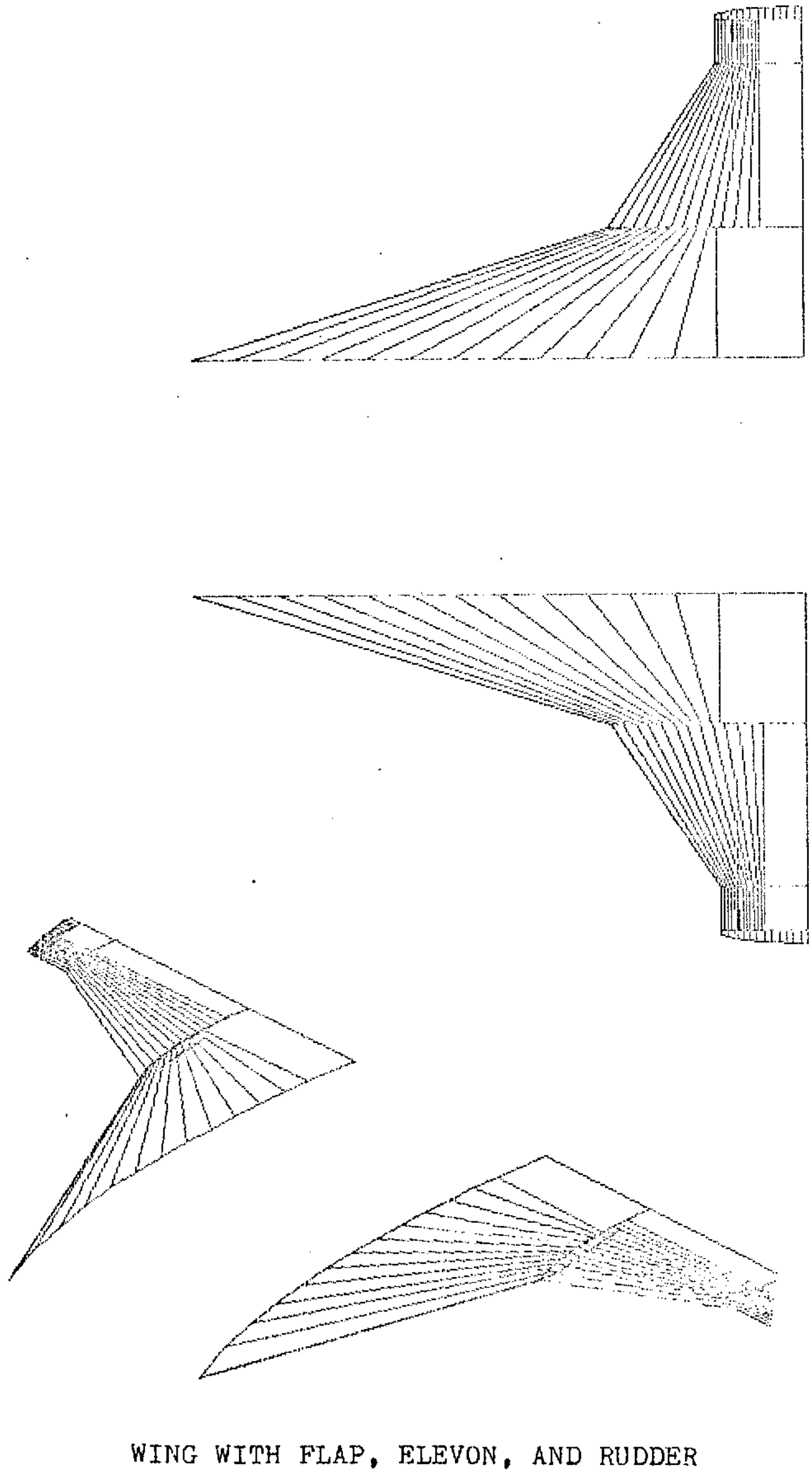
```
?EXECUTE GTM≠
C. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.  BUILD ELEVON AND RUDDER
C. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
SECTION EDIT
C.
C.  ELEVON IS BUILT FROM CRANK AND TIP
C.  RUDDER IS BUILT FROM ELEVON TIP AND VT1
C.
     ACCESS 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.
C.  ELEVON AND RUDDER ARE ON THE NEGATIVE SIDE OF THE CUTTING PLANE
C.  THE REDEFINED WING IS ON THE POSATIVE SIDE
C.
      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 UT
     ACCESS RUDDER UT
     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
```


GTM INPUT--ELEVON AND RUDDER SURFACES

```
C.
C.   INSURE WING SEGMENTS HAVE 13 POINTS
C.
       ACCESS CRANKO UPPER
       NSEG = 12
       EQARC
       ACCESS CRANKO LOWER
       NSEG = 12
       EQARC
       ACCESS TIP UPPER
       NSEG = 12
       EQARC
       ACCESS TIP   LOWER
       NSEG = 12
       EQARC
       ACCESS VT1 UPPER
       NSEG = 12
       EQARC
       ACCESS VT1 LOWER
       NSEG = ØØ
       EQARC
C.
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. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.  BUILD ELEVON COMPONENT
C. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      ACCESS VEHICLE ORBITER
      BUILD = ACCESS
      BUILD COMPONENT ELEVON UPPER
      BUILD  SECTION ROOT
     COPY SEGMENT ELEVON UR
     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. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
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 TIP
     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
```



WING WITH FLAP, ELEVON, AND RUDDER

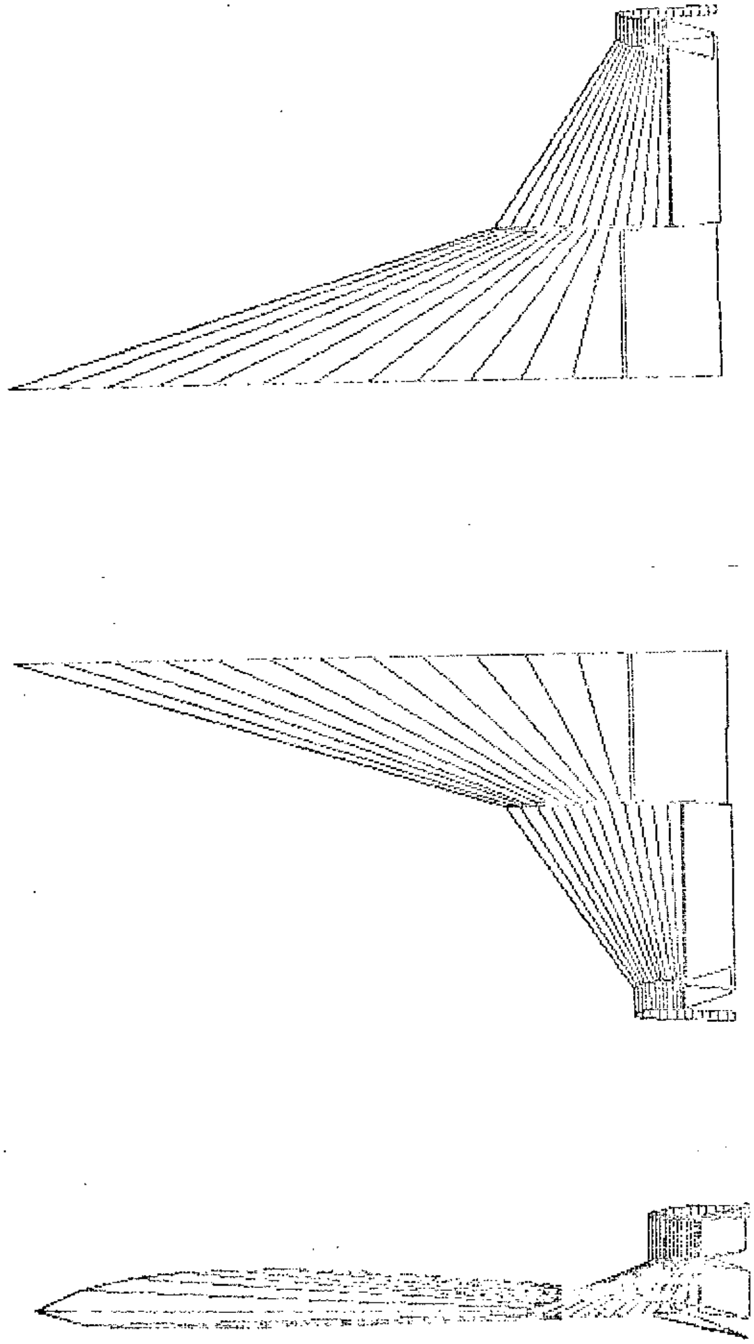
Step 5: Rotations and Deflections

The preceeding four steps created the desired geometry in a flat wing. This step accomplishes the rotations such that the vertical control surfaces have a dyhedral of 60°, and the outboard wing section has a dyhedral 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:

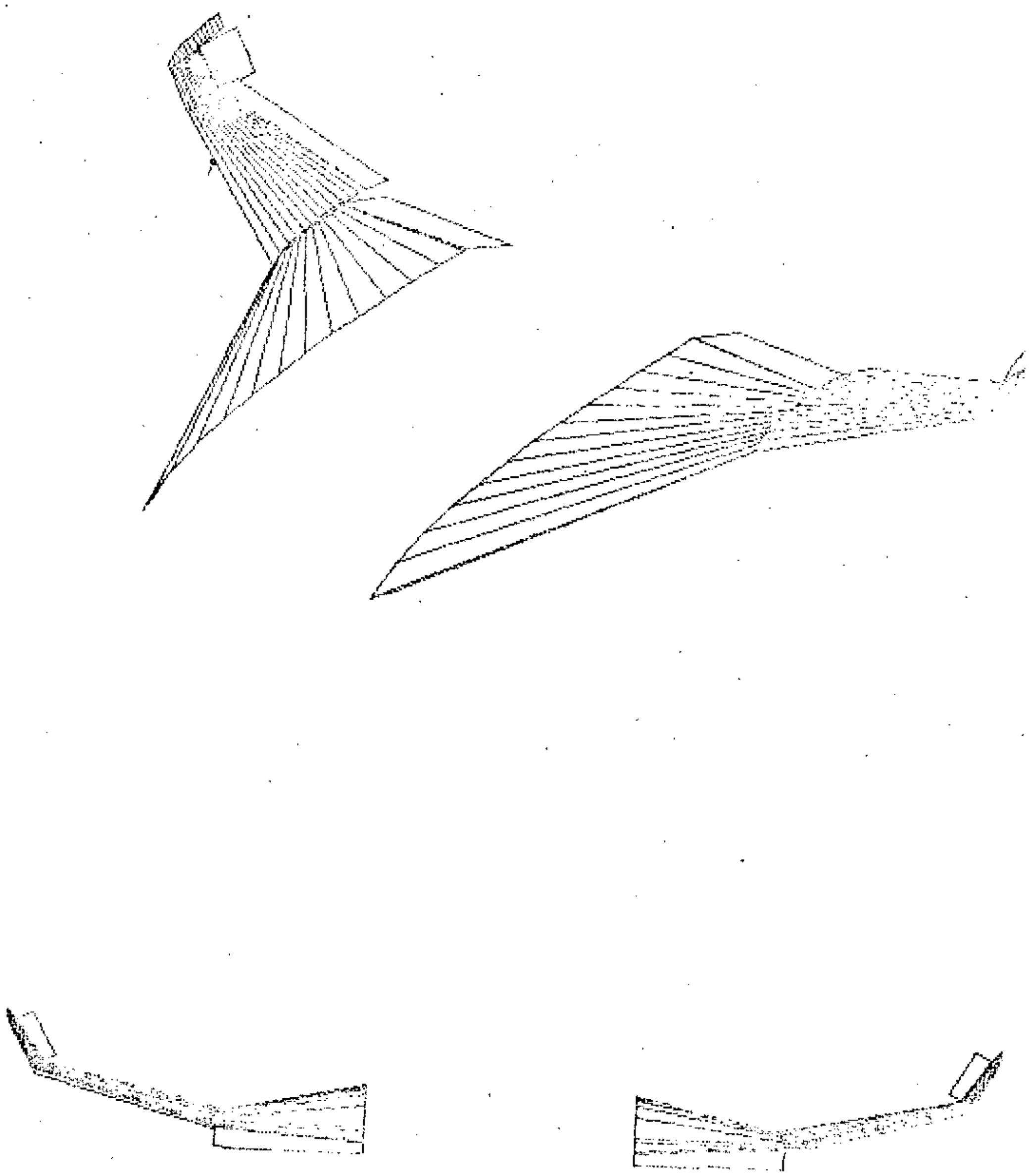
```
≠EXECUTE GTM≠
C. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C.  ROTATE WING INTO THE DESIRED SHAPE
C. - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
SECTION EDIT
      ACCESS VEHICLE ORBITER
C.
C.  RUDDER DEFLECTION ROTATION
C.
      ACCESS COMPONENT RUDDER UPPER
     RCEN = -2576,0,-132
     PITCH = -25
     ROT
     ACCESS COMPONENT RUDDER LOWER
     ROT
C.
C.  ELEVON DEFLECTION ROTATION
C.
     PITCH = 10
      ACCESS COMPONENT ELEVON UPPER
     ROT
      ACCESS COMPONENT ELEVON LOWER
      ROT
C.
C.  FLAP DEFLECTION ROTATION
C.
     RCEN = -2476  ,0,-132
     PITCH = 15
      ACCESS COMPONENT FLAP UPPER
     ROT
      ACCESS COMPONENT FLAP LOWER
     ROT
C.
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
```

```
C.
C.  SECOND WING ROTATION--ALL OUTBORD OF CRANK UP 15 DEGREES
C.
     RCEN = -2676,573,-132
     ROLL = 15
     ROLL =-15
      ACCESS COMPONENT RUDDER UPPER
    ROT
     ACCESS COMPONENT RUDDER LOWER
     ROT
     ACCESS COMPONENT ELEVON UPPER
     ROT
     ACCESS COMPONENT ELEVON LOWER
     ROT
      ACCESS COMPONENT  WING UPPER
     START = CRANKO UPPER
     ROT
      ACCESS COMPONENT WING LOWER
      STOP = CRANKO LOWER
     ROT
     TREE LIST ORBITER
     OUTPUT ORBITER
EXIT
     SAVE DATA BASE
STOP
*EOF
```

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 accomodate 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:

Station-x	Magnification Factor
-1	.0166
-5	.0435
-10	.0659
-20	.0999
-50	.1731
-100	.2624
-150	.3333
-200	.3977
-500	.6891

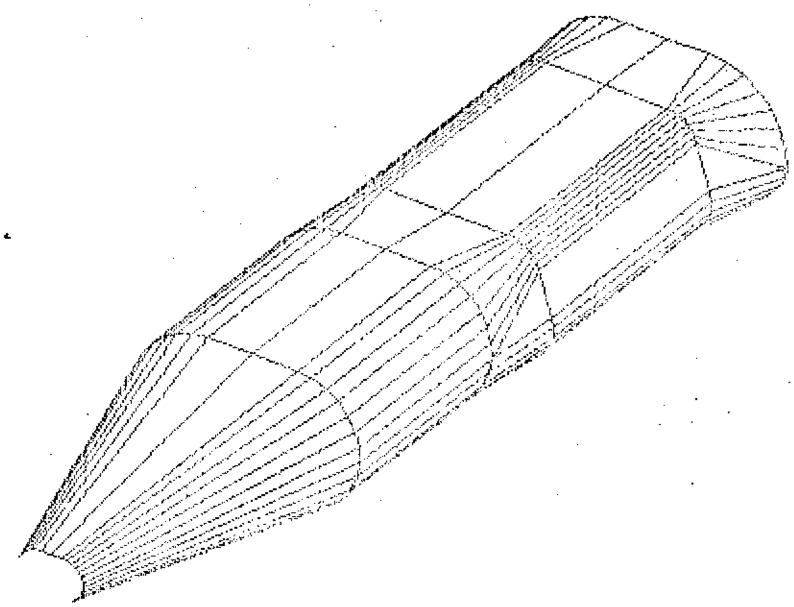
The command sequence for the GTM used for this enhancement is as follows:

```
#EXECUTE GTMA
IMAGE INPUT
      GEOTRY GEOMETRY
       DATA BASE ARC BODY
      V=ARC BODY
      C=BODY
      C=A
      C=[illegible]
      C=C
      EXIT
      EXIT
      EXIT
VEHICLE EDIT
      THEE LIST ARC BODY
      ACCESS VEHICLE ARC BODY
      ACCESS COMPONENT BODY
      BUILD SEGMENT BASE
      COPY SECTION SECTION 2
      EXIT
SEGMENT EDIT
      L BASE
       MAG = 1
      XMAG = [illegible]
      SC BASE
      BUILD SEGMENT NEW [illegible]
      MAG = .0144
      SC BASE
      XMOVE = -1
      MOVE
      BUILD SEGMENT NEW [illegible]
      MAG = .0435
      SC BASE
      XMOVE = -5
      MOVE
      BUILD SEGMENT NEW [illegible]
      MAG = .0650
      SC BASE
      XMOVE = -10
      MOVE
      BUILD SEGMENT NEW [illegible]
      MAG = .1000
      SC BASE
      XMOVE = -20
      MOVE
      BUILD SEGMENT NEW [illegible]
      MAG = .1721
      SC BASE
      XMOVE = -50
      MOVE
```

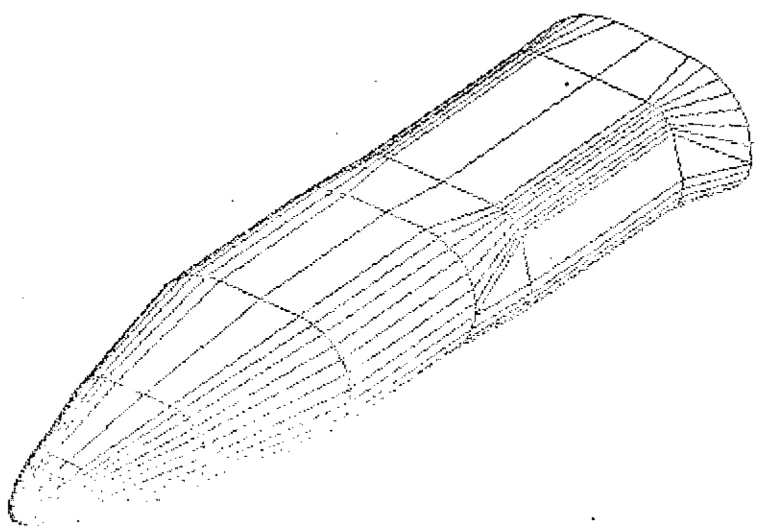
GTM INPUT -- BODY ENHANCEMENT

```
        BUILD SEGMENT NEW 6
        MAG = .2624
        SC BASE
        XMOVE = -100
        MOVE
        BUILD SEGMENT NEW 7
        MAG = .3977
        SC BASE
        XMOVE = -200
        MOVE
        BUILD SEGMENT NEW 8
        MAG = .6891
        SC BASE
        XMOVE = -500
        MOVE
        BUILD SEGMENT NEW 9
        MAG = .333
         SC BASE
        XMOVE = -150
        MOVE
EXIT
VEHICLE EDIT
        ACCESS VEHICLE ARC BODY
        ACCESS COMPONENT 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 2
        INSERT SECTION NEW 7
        ACCESS SECTION SECTION 2
        INSERT SECTION NEW 8
        ACCESS SECTION SECTION 1
         REPLACE SECTION NEW 9
        TREE LIST ARC BODY
        OUTPUT ARC BODY
        EXIT
SAVE DATA BASE
STOP
#EOF
```

GTM INPUT -- BODY ENHANCEMENT (cont.)



ORIGINAL MINIMUM ENVELOPE BODY



ENHANCED MINIMUM ENVELOPE BODY