General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)
IMP: INTERACTIVE MASS PROPERTIES PROGRAM
VOLUME I - PROGRAM DESCRIPTION

By: William A. Stewart

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Space Center
Houston, Texas 77058

June 1976
PREFACE

This report was prepared under Contract NAS9-14520, The Development of the Engineering Design Integration (EDIN) System, A Computer Aided Design Development Effort. The contract was funded by the National Aeronautics and Space Administration, Johnson Space Center, Engineering Analysis Division. Mr. Robert W. Abel was the contract monitor.
CONTENTS

SUMMARY .............................................................................................................. 1

INTRODUCTION ...................................................................................................... 3

VOLUME, AREA AND MASS PROPERTIES ANALYSIS ........................................... 5

Surface Model .......................................................................................................... 6

Mass Properties of a Triangular Surface Element .................................................. 8

Transformation Matrix for Triangle Elements ...................................................... 10

Mass Properties of the Surface ............................................................................ 13

Area Calculations ................................................................................................... 14

Volume Calculations .............................................................................................. 14

Black Box Contributions to the Mass Properties ............................................... 15

REFERENCES ......................................................................................................... 17

APPENDIX A - SUBROUTINE DESCRIPTIONS ..................................................... 19
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Area</td>
</tr>
<tr>
<td>A&lt;sub&gt;p&lt;/sub&gt;</td>
<td>Projected Area</td>
</tr>
<tr>
<td>B</td>
<td>Base of Triangular Element</td>
</tr>
<tr>
<td>B&lt;sub&gt;X&lt;/sub&gt;</td>
<td>X-Projection of the Base of the Triangular Element</td>
</tr>
<tr>
<td>B&lt;sub&gt;Y&lt;/sub&gt;</td>
<td>Y-Projection of the Base of the Triangular Element</td>
</tr>
<tr>
<td>B&lt;sub&gt;Z&lt;/sub&gt;</td>
<td>Z-Projection of the Base of the Triangular Element</td>
</tr>
<tr>
<td>C</td>
<td>Offset of the Vertex of a Triangular Surface Element</td>
</tr>
<tr>
<td>CV</td>
<td>Center of Volume</td>
</tr>
<tr>
<td>D</td>
<td>One side of a Triangular Surface Element</td>
</tr>
<tr>
<td>D&lt;sub&gt;C&lt;/sub&gt;</td>
<td>Matrix of Direction Cosines of a Triangular Surface Element</td>
</tr>
<tr>
<td>D&lt;sub&gt;X&lt;/sub&gt;</td>
<td>X-Projection of the D-Side of a Triangular Surface Element</td>
</tr>
<tr>
<td>D&lt;sub&gt;Y&lt;/sub&gt;</td>
<td>Y-Projection of the D-Side of a Triangular Surface Element</td>
</tr>
<tr>
<td>D&lt;sub&gt;Z&lt;/sub&gt;</td>
<td>Z-Projection of the D-Side of a Triangular Surface Element</td>
</tr>
<tr>
<td>H</td>
<td>Thickness of a Triangular Surface Element</td>
</tr>
<tr>
<td>I</td>
<td>Element Index in a Section Plane</td>
</tr>
<tr>
<td>J</td>
<td>Element Index of Sections</td>
</tr>
<tr>
<td>I&lt;sub&gt;xx&lt;/sub&gt;</td>
<td>Moment of Inertia (X-Axis)</td>
</tr>
<tr>
<td>I&lt;sub&gt;yy&lt;/sub&gt;</td>
<td>Moment of Inertia (Y-Axis)</td>
</tr>
<tr>
<td>I&lt;sub&gt;zz&lt;/sub&gt;</td>
<td>Moment of Inertia (Z-Axis)</td>
</tr>
<tr>
<td>I&lt;sub&gt;xy&lt;/sub&gt;</td>
<td>Product of Inertia (X-Y Plane)</td>
</tr>
<tr>
<td>I&lt;sub&gt;yz&lt;/sub&gt;</td>
<td>Product of Inertia (Y-Z Plane)</td>
</tr>
</tbody>
</table>
Product of Inertia (Z-X Plane)

Inertia Matrix Centered at a Triangular Element c.g.

Inertia Matrix Centered at the Reference System Axes.

Enclosed Volume of Elements.

Measure of Distance in the X Direction

X Coordinate of the Jth Section

Y Coordinate of the Base of the Triangular Element

Component of Unit Vector in the DC Matrix

Component of Unit Vector in the DC Matrix

Component of Unit Vector in the DC Matrix

Local (Triangle) x-Axis

Measure of Distance in the Y Direction

Y Coordinate of (I, J) Element

Y Coordinate of the Base of the Triangular Element

Component of Unit Vector in the DC Matrix

Component of Unit Vector in the DC Matrix

Component of Unit Vector in the DC Matrix

Local (Triangle) y-Axis

System Reference Axis

Z-Coordinate of (I, J) Element

Z-Coordinate of the Base of the Triangular Element

Component of Unit Vector in the DC Matrix

Component of Unit Vector in the DC Matrix

Component of Unit Vector in the DC Matrix

Local (Triangle) y-Axis
IMP: INTERACTIVE MASS PROPERTIES PROGRAM
Volume I - Program Description
by
William A. Stewart - Sigma Corporation

SUMMARY

This document describes a method of computing a weights and center of gravity analysis of a flight vehicle using interactive graphical capabilities of the Adage 340 computer. The equations used to calculate area, volume and mass properties are the equations derived in reference 1. The input/output methods are different than those presented in reference 1 because of the graphic support of the Adage computer and the geometry definition used for storing data. Several interactive program options are available for analyzing the mass properties of a vehicle. These options are explained in this report. An overall picture of the program with its displays and options is shown in figure 1.

The document also describes the method and computer techniques for the calculation of the mass properties, c.g. location, enclosed volume, wetted area and planform area of any closed structure that has a plane of symmetry. The vehicle is described to the computer program by ordered sets of X, Y, Z coordinates of points on its surface. The X, Y Z coordinates are converted to quadrilateral elements for analysis. The mass properties of each quadrilateral may be computed from a thickness and density input for each quadrilateral or from a weight per unit area input at each point or from a combination of both. The weight per unit area can be composite of the real wall including skin, insulation, ribs, stringers, standoffs, brackets, etc.

The mass properties of the quadrilateral elements are accumulated for each section of the vehicle and for the total vehicle. The computed mass property totals may contain not only the contribution from the distributed mass on the vehicle surface wall, but additional masses may be added as "black boxes" by specifying each one's c.g. location and mass properties. The added black boxes may be inside or outside the surface and do not have to be symmetrical with respect to X-Z plane.
FIGURE 1  INTERACTIVE MASS PROPERTIES PROGRAM.
INTRODUCTION

The estimation of the mass properties of a vehicle is one of the most important considerations in the design process and yet one of the most inexact engineering endeavors. While the calculation of aerodynamic, propulsion and mission performance are based on widely recognized mathematical prediction techniques, the estimation of weight must be based largely on historical data. The art of weight estimation has evolved through the years by the diligent collection and correlation of component weights of previously built vehicles. New design weights are predicted on the basis of the component weights of past designs. Little information is usually available on the other properties such as volume, area, center of gravity and inertia of the components. Even for "weights only" calculation, historical methods lack validity in predicting advanced state-of-the-art design data. Further, the influence of small design perturbations eludes such gross prediction methods.

During the later design phases when detail structured analyses are available for some components, the weights engineer must still revert to empirical analysis methods for subsystem and secondary structure weights such as wing leading edges, canopies, fairings, etc.

Existing programs such as references 2 through 6 lean heavily on the historical approach for weight estimating. To obtain more reliable information, the designer must turn to structural analysis and subsystem design programs such as references 7 through 13. The detailed geometry descriptions required for these classes of analysis are usually not known until the later design phases. At this later phase the design is usually frozen and little geometric perturbation is allowed. Further, small perturbation analysis using these programs is very expensive in terms of manpower and computer resources.

Although the precise weight and other properties of the vehicle components are illusive, or at best a very complex calculation, mathematical equations for combining the components mass properties into total vehicle mass properties are quite concise. Mass property evaluations for the total vehicle as well as perturbations due to added or deleted masses are easily determined once the mass properties of all the components are known (or assigned).
The significant contribution of the IMP program development to the general problem of computer aided design is threefold. First, the program offers a unified approach to the accounting of the vehicle component mass properties. The fact that a component weight is not well known does not exclude it from being analyzed in a mass properties evaluation. Furthermore, detailed mass properties of some vehicle components may be determined elsewhere and still be included in the IMP analysis. Second, the distributed mass surface model incorporated in IMP allows a very convenient means of producing vehicle geometric perturbations early in the design process with some degree of confidence in the relative effect of the perturbation. As more detailed information on the design becomes available, the component input to IMP may be gradually replaced without distribution of the overall continuity of the weights analysis. Third, the IMP program allows the user to directly interact with the mass properties evaluation process through use of the Adage 340 computer. This highly sophisticated display computer allows the user an additional dimension in his analytical capabilities.
The mathematical modeling of the IMP program permits the calculation of the mass properties, c.g. locations, enclosed volume, wetted area and planform area of any closed structure that has a plane of symmetry, c.g. fuselage, stiffened fuel tank, etc. The vehicle is described in a symmetrical manner and consists of sets of X, Y, Z coordinates on its surface. The surface coordinates are converted to quadrilateral elements but since the four corners of the quadrilaterals are not necessarily coplanar, each quadrilateral is analyzed as two triangles. The mass properties are computed from a thickness and density input for each quadrilateral, from a weight per unit area input at each point, or from a combination of both. The weight per unit area can be a composite of the real wall including skin, insulation, ribs, stringers, standoffs, brackets, etc.

The elemental mass properties are accumulated for each section and for the total vehicle. The computed mass property totals containing the contribution from the distributed mass in the vehicle surface wall may be combined with additional masses specifying each one's c.g. location and mass properties about its c.g. The added black boxes may be inside or outside the surface and do not have to be symmetrical.

The properties of the vehicle computed by the program are:

1. Weight.
2. Coordinates of the c.g.
3. Three mass moments of inertia plus three products of inertias.
4. Surface area.
5. Enclosed volume.
6. Projected area.

These properties are output for each segment along with cumulative totals and complete vehicle values.
The numerical model of the surface shape to be analyzed is described by quadrilateral elements. Each quadrilateral element consists of a grouping of four surface points. An organized set of quadrilateral elements form a segment of the configuration. A number of segments may be used to provide a complete description of the configuration.

The configuration is usually (but not necessarily) positioned with its nose at the coordinate system origin and with the length of the body stretching in a positive X-direction. The coordinate system used for this program is the right hand Cartesian system as illustrated below:

The configuration is assumed symmetrical about the X-Z plane, and the analysis accounts for the plane of symmetry.

The configuration may consist of an arbitrary number of wing and body sections. The body sections are described in the positive half plane perpendicular to the plane of symmetry. The wings are described as airfoil sections parallel to the plane of symmetry. Canards and horizontal stabilizers are described similar to wings. Fins cannot be described as airfoil sections but they may be described as body sections.
Points along the X-axis are referred to as station coordinates. A Y-Z plane is a station plane. The Y-Z coordinates in the station plane form a station contour.

A portion of the mathematical surface model is presented in the illustration above. The X,Y,Z coordinates of the points where the solid grid lines cross are described to the program. Each point (I,J) is associated with a station number J and a contour point I. Thus, the circled point is 8,5 and it has coordinates X5;Y8;Z8. The model
groups the surface points into quadrilateral elements which also have unique I,J identification. The crosshatched one is 8,5 corresponding to its lowest numbered corner. The dashed diagonals illustrate the division of each quadrilateral into two triangles for analysis.

Airfoil sections are described in a similar manner as the body sections above. In the case of airfoil section, the parts are described in X-Z planes parallel to the plane of symmetry.

The objective in the analysis is to calculate the properties of each quadrilateral (two triangles) elements and transform the results back to the reference coordinate system, then accumulate the properties from all elements of the surface.

Mass Properties of a Triangular Surface Element. - The triangular surface element illustrated below with its base a station J and apex at J+1 is assumed to be of uniform thickness and homogeneous.

As viewed from the outside of the vehicle, the local x,y,z coordinate system has its origin at (I,J). The x-y plane is coincidental with the plane of the triangle. A is the height of the triangle. B is the base of the triangle.
C is the offset of the apex from the y-axis. H is the thickness which is assumed to be small compared to A, B and C.

The area, $S$ of this elemental triangle is:

$$S = \frac{AB}{2} \quad (1)$$

The first moments of inertia for the same triangle are:

$$S_x = \frac{\rho HS(B+C)}{2} \quad (2)$$
$$S_y = \frac{\rho HSA}{2} \quad (3)$$
$$S_z = 0 \quad (4)$$

The second moments of inertia for the elemental triangle are:

$$I_{xx} = \frac{\rho HA^2 S}{6} \quad (5)$$
$$I_{yy} = \frac{\rho (B^2 + BC + C^2)S}{6} \quad (6)$$
$$I_{zz} = I_{xx} + I_{yy} \quad (7)$$

Where $\rho$ is the density of the thin triangle. Finally, the products of inertia are:

$$I_{xy} = \frac{\rho H (B+2C)AS}{12} \quad (8)$$
$$I_{yz} = 0 \quad (9)$$
$$I_{xz} = 0 \quad (10)$$

The coordinates of the centroid (center of gravity) of the triangular element are:

$$\bar{x} = \frac{(B+C)}{3} \quad (11)$$
$$\bar{y} = \frac{A}{3} \quad (12)$$
The transformation of the inertias and products of inertia to the centroid yield:

\[ I_{xx} = \frac{SA^2}{18} \quad (13) \]
\[ I_{yy} = \frac{X(B^2-BC+C^2)}{18} \quad (14) \]
\[ I_{xy} = \frac{SA(ZC-B)}{36} \quad (15) \]
\[ I_{zz} = I_{xx} + I_{yy} \quad (16) \]

Transformation Matrix for Triangle Elements. - The mass properties above are given for a simple triangular element in the local x,y,z coordinate system. The elemental mass properties must be transformed to the reference coordinate system X,Y,Z for cummulation with other elements. The transformation from the primary x,y,z coordinate system to the local system is given by:

\[
\begin{bmatrix}
  x \\
  y \\
  z
\end{bmatrix} = [DC]
\begin{bmatrix}
  X - XO \\
  Y - YO \\
  Z - ZO
\end{bmatrix} \quad (17)
\]

Where XO,YO,ZO are the coordinates of the prime system origin with respect to same arbitrary reference point and DC is a matrix of direction cosines. The elements of DC are:

\[
[DC] = \begin{bmatrix}
  XPX & XPY & XPZ \\
  YPX & YPY & YPZ \\
  ZPX & ZPY & ZPZ
\end{bmatrix} \quad (18)
\]

For a unit vector in the x direction, XPX is the projection of x on X, XPY is the projector of x on Y, etc. The vertices of the triangular element in local x,y,z coordinates may be considered as vertices B and D, drawn from the origin (0,0,0). The components of B and D are developed as follows. That is, BX is the X component of B and DY is the Y component of D, etc. Therefore:
If: 
\begin{align*}
X_O &= X(J) \\
Y_O &= Y(I,J) \\
Z_O &= Z(I,J)
\end{align*}  \hspace{1cm} (19)

Then: 
\begin{align*}
B_X &= 0.0 \text{ (B lies in YZ Plane)} \hspace{1cm} (22)
B_Y &= Y(I+1,J) - Y_O \hspace{1cm} (23)
B_Z &= Z(I+1,J) - Z_O \hspace{1cm} (24)
D_X &= X(J+1) - X_O \hspace{1cm} (25)
D_Y &= Y(I+1,J+1) - Y_O \hspace{1cm} (26)
D_Z &= Z(I-1,J-1) - Z_O \hspace{1cm} (27)
\end{align*}

The cross product of B with D is a vector \( \vec{S^2} \) in the z direction. Its magnitude is twice the surface area of the triangle. Let \( S_X, S_Y \) and \( S_Z \) be the components of \( S^2 \). Since \( B_X = 0 \), the cross product yields:
\begin{align*}
S_X &= B_Y*D_Z - D_Y*B_Z \hspace{1cm} (28) \\
S_Y &= B_Z*D_X \hspace{1cm} (29) \\
S_Z &= -D_X*B_Y \hspace{1cm} (30)
\end{align*}

Let \( B, D, S^2 \) represent the magnitude of their respective vectors. Then \( B/B = \) unit vector in x direction and \( S^2/S^2 = \) unit vector in z direction.

It follows:
\[ \frac{S^2}{S^2} \text{ cross } \frac{B}{B} = \text{ unit vector in y direction} \hspace{1cm} (31) \]

The components of each unit vector are a row in the matrix DC. Therefore:
\begin{align*}
X_PX &= BX/B = 0.0 \hspace{1cm} (32) \\
X_PY &= BY/B \hspace{1cm} (33) \\
X_PZ &= BZ/B \hspace{1cm} (34) \\
Z_PX &= SX/S^2 \hspace{1cm} (35) \\
Z_PY &= SY/S^2 \hspace{1cm} (36) \\
Z_PZ &= SZ/S^2 \hspace{1cm} (37) \\
YPX &= ZPX*X_PZ - X_PY*Z_PZ = DX/A \hspace{1cm} (38)
\end{align*}
YPY = XPZ*ZPX \hspace{1cm} (39) 
YPZ = ZPX*XPY \hspace{1cm} (40) 

The preceding steps considered a triangle with its base on J and apex on J+1. The other half of each quadrilateral has its base on J+1 and vertex on J. The vertices are identified as follows:

The analysis for the inverted triangle is similar to the upright triangle. The only difference is in equations (13) through (21). For the inverted triangle,

\[ 
XO = X(J+1) \hspace{1cm} (41) 
\]
\[ 
YO = Y(I+1,J+1) \hspace{1cm} (42) 
\]
\[ 
ZO = Z(I+1,J+1) \hspace{1cm} (43) 
\]
\[ 
BX = 0.0 \hspace{1cm} (44) 
\]
\[ 
BY = Y(I,J+1) - YO \hspace{1cm} (45) 
\]
\[ 
BZ = Z(I,J+1) - ZO \hspace{1cm} (46) 
\]
\[ 
DX = X(J) - XO \hspace{1cm} (47) 
\]
\[ 
DY = Y(I,J) - YO \hspace{1cm} (48) 
\]
\[ 
DZ = Z(I,J) - ZO \hspace{1cm} (49) 
\]

The above analysis is similar for airfoil type sections described in the x-z plane.
Mass Properties of the Surface. - The c.g. and mass properties expressions for a triangle are given in the previous section. The triangular element parameters $A$ and $B$ have been developed above and are always positive. Now $C^2 = D^2 - A^2$ but $C$ can be negative. However, $C$ and its sign are given directly by the dot product of $D$ with the unit vector in the $x$ direction.

$$C = DY*XPY + DZ*XPZ$$  (50)

The resulting c.g. coordinates are in the $x$, $y$, $z$ system and the inertias are about axes through the c.g., parallel to the prime system. They are transformed to the system reference axes as follows:

$$\begin{bmatrix}
XBR \\
YBR \\
ZBR
\end{bmatrix} = \begin{bmatrix}
XO \\
YO \\
ZO
\end{bmatrix} + [DC]^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$  (51)

This is readily expanded since the inverse of $DC$ is identical to its transpose. The products and moments of inertia parallel to the prime axes are assembled in $PMIP$:

$$PMIP = \begin{bmatrix}
I_{XX} & -I_{XY} & -I_{XZ} \\
-I_{XY} & I_{YY} & -I_{YX} \\
-I_{XZ} & -I_{YZ} & I_{ZZ}
\end{bmatrix}$$  (52)

The minus signs on the products of inertia are a consequence of the coordinate system definition. This places the matrix $PMIP$ in tensor form. It can be shown that the inertias are rotated parallel to the system axes by:

$$[PMI] = [DC]^T \ast [PMIP] \ast [DC]$$  (53)

$PMI$ contains the inertias about axes parallel to the system reference axes, but centered on the triangle's c.g. Translation to the system origin makes use of the parallel axis theorem along with $XBR$, $YBR$, $ZBR$ computed above, and is included with the summation procedure described in the next paragraph. The products of inertia contained in $PMI$ are the negative of their engineering definition, like they are in $PMIP$. 
In the IMP analysis, all values in PMI are twice the true value since $S_2 = \text{twice the area of the triangle, } S_2$ has been retained through all the computations. But each triangle has a symmetrical counterpart on the other side of $Y = 0$ whose contribution to the total is either equal or equal-and-opposite. In the latter case, the resultant is zero; i.e., $I_{XY} = I_{YZ} = 0.0$.

**Area Calculations.** - The surface area and projected area result from the summation of the elemental areas from the above analyses. The surface area is:

$$A = \frac{1}{2} \sum S_{2i}$$  \hspace{1cm} (54)

$S_2$ is twice the elemental area and the projected area (on the $y$-$z$ plane) is:

$$A_p = \frac{1}{2} \sum S_{Yi}$$  \hspace{1cm} (55)

**Volume Calculations.** - The elemental volume of a surface element is the projection of the elemental area on the $x$-$y$ plane multiplied by the average distance to the c.g. of the element. The total volume is:

$$V = \sum x_i \cdot ZP_{Xi} \cdot A_i$$  \hspace{1cm} (56)

The sign of the direction cosine accounts for whether the elemental volume is added or subtracted.

The center of volume is simply:

$$CV = \frac{1}{V} \sum x_i^2 \cdot ZP_{Xi} \cdot A_i$$

The above calculations are approximate because of the assumption that the c.g. of the element is an adequate approximation for the entire elemental surface. The assumption is reasonable for small elements but may not be valid for a gross panelling of the configuration.
Black Box Contributions to the Mass Properties

The surface model described above is suitable for describing shell structures such as tanks, skins, etc. However, the addition of mass not conforming the above model may be necessary in the complete description of the vehicle mass properties. In the IMP analysis, additional mass sources may be added by specifying each one's center of gravity (c.g.) location and mass properties. These mass sources called "black boxes" may lie inside or outside the surface and do not have to be symmetrical with respect to the x-z plane. The IMP program combines the detailed shape inputs and the black box data to produce the overall mass properties of the vehicle.

Black box masses are described to the program in the following manner:

1. Location of its c.g. in system reference coordinates.
2. Total weight.
3. Moments and products of inertia about local axes through its c.g.
4. Orientation of local axes relative to the system reference axes or relative to some intermediate axes.

The minimum required for a black box input is the specification of items 1 and 2. If items 3 and 4 are omitted, the moments and products of inertia are assumed to be zero, and the orientation of the local axis is assumed to be coincidental with the system reference axes. The contribution of each black box is added to the cumulative totals produced by the analysis. Since the size and shape is not being specified, there is no addition to surface area or volume. Since the black boxes do not have to be symmetrical with respect to the x-z plane, $I_{XY} \neq I_{YZ}$ and there can be a non-zero first moment with respect to $Y$.

The moments and products of inertia of each black box are input directly to PMIP (see equation 52). The program takes care of filling the upper half and adding the minus
signs. Rotation of the inertias uses equation 53. DC is obtained from input data as follows. Consider the local x,y plane:

![Diagram of x, y plane with points 0, 1, 2]

Point 0 is the origin. Points 1 and 2 are arbitrary points on the +x and +y axes respectively. Coordinates of these three points, measured in the system to which the inertias are being rotated, are inputs. It is a simple matter to construct unit vectors in the x and y directions and their components yield the first two rows in DC, equation 18. The third row is the cross product of these two unit vectors.

In order to construct DC, it is not necessary that point 0 be the c.g. The coordinates x, y must be parallel to the axes to which the PMIP inputs are referenced, but rotation is performed independently of translation. After the rotation of equation 53, the inertias in PMI are still centered on the c.g. of the black box. Only the rotation takes place and the contents of PMI are moved to PMIP. Thus, it is possible to make several rotations, if needed, to reach alignment with the system reference axes. The input to the program can be developed from any convenient points on the black box and intermediate structure whose coordinates are known. The input does not always have to be in the same units as the system X,Y,Z data.
REFERENCES


APPENDIX A

SUBROUTINE DESCRIPTIONS

This appendix describes the subroutines used by the interactive mass properties program and their general relationship to the overall functions of the IMP program. Program MAIN is the main driver routine. Subroutine listings follow the descriptions below.

Subroutine ANRST

ANRST is a Fortran subroutine which initializes variables used in the subroutines AN6 and AN7. This subroutine is internal to AN6 and AN7.

Subroutine AN6

AN6 is a Fortran subroutine that generates an input number within the input grid.

Subroutine AN7

AN7 is a Fortran subroutine that creates an input number within the input grid. This subroutine is used exclusively by WABCOM. The routine differs from AN6 in that AN6 is used by all other subroutines.

Subroutine APNDF

APNDF is an assembly language subroutine which allows the user to append data records from core to the end of a disk file within a pack/volume. The subroutine is part of a general file access package called RANDOM IO.

Image BKBX

BKBX is an image subroutine written in Fortran which displays the mass properties of a black box and allows the user to make changes to the black box data interactively. This image is internal to subroutine WABCOM.

Subroutine BLDMAN

BLDMAN is a Fortran subroutine that converts an ATEXT file name to a format that can be used by the RANDOM IO routines.

Subroutine CGMOVE

CGMOVE is a Fortran subroutine that supports the black box tracker option of the program. It allows the user to translate components with respect to a fixed fuselage.
Subroutine CLOSF

CLOSF is an assembly language subroutine which supports the RANDOM IO package. The subroutine is used to close a file which has been opened by the STRTF subroutine and was filled by calls to APNDF.

Subroutine CROSS

CROSS is a Fortran subroutine that computes the cross product of two vectors.

Subroutine DECFMT

DECFMT is a Fortran subroutine that computes an output ATEXT format for an internal binary decimal number. The routine is used in conjunction with menu generation.

Subroutine DSFIL

DSFIL is an assembly language routine which is part of the RANDOM IO package. The routine checks for validity or duplication of a file name on a given pack/volume. If the file is found to already exist, the size and disk address are returned from the subroutine.

Subroutine DSKER

DSKER is an assembly language subroutine that supports the RANDOM IO package. DSKER is an assembly language subroutine that provides a testing loop and restricts further operations until all requested disk operations are complete, and the disk and controller are ready for the next request. In general, the routine controls the timing within the RANDOM IO package.

Subroutine DSVOL

DSVOL is an assembly language subroutine that checks the validity and free space of a pack/volume. This subroutine supports the RANDOM IO package.

IMAGE GEO

GEO is a Fortran IMAGE subroutine that displays the input values for the geometry calculations in the WABCOM subroutine. This subroutine allows the user to make changes to these inputs interactively. This image is internal to subroutine WABCOM.
IMAGE GRID

GRID is an IMAGE that displays an alpha numeric grid that allows the user to make inputs interactively.

Subroutine INPUT

INPUT allows the user to construct a data file that is used interactively with the mass properties program.

FUNCTION IOCT

IOCT is a Fortran function that converts the decimal number to its octal equivalent.

Subroutine LOAD

LOAD is a Fortran subroutine that sets up array values so that they are properly used in the calculation of the WABCOM subroutine.

Subroutine LOCFLE

LOCFLE is a Fortran subroutine that calculates the file sequence number within a pack/volume. This is a subroutine that takes a file name and locates its position within a pack/volume by its sequence number.

Program MAIN

This program is the executive control program for the IMP program.

Subroutine MSM3X3

This subroutine performs the multiplication of a three by three (3X3) matrix by another three by three (3X3) matrix.

Subroutine MXD

MXD is a Fortran subroutine that performs the multiplication of a three by three (3X3) matrix by one by three (1X3) matrix.

IMAGE OPTIONS

OPTIONS is a Fortran image subroutine that displays the various options used in the interactive mass properties program. This image subroutine allows the user to choose the option that he wishes.
Subroutine PAT

PAT is a Fortran subroutine that performs the calculation of the parallel axis theorem. The parallel axis theorem is a method of summing center of gravities and inertias of several component parts to achieve the total values for the system.

Subroutine RDSKOK

RDSKOK is a Fortran language subroutine that verifies the existence of a pack/volume and file on disk. If successful, the number of words and the beginning address are returned.

Subroutine RESET

RESET is a Fortran subroutine that initializes variables in the WABCOM subroutine. This subroutine is internal to WABCOM.

Subroutine RESET1

RESET1 is a Fortran subroutine that initializes variables in the interactive mass properties program. This subroutine is internal to program MAIN.

Subroutine RESET2

RESET2 is a Fortran subroutine that initializes variables in the interactive mass properties program. This subroutine is internal to program MAIN.

Subroutine RESET69

RESET69 is a Fortran subroutine that initializes variables in the weights and balance subroutine. This subroutine is internal to WABCOM.

Subroutine RNDRD

RNDRD is an assembly language subroutine that randomly reads files access by the RANDOM IO package.

Subroutine SKETCH

SKETCH is a Fortran subroutine that calculates the coordinates of the fuselage used in the black box tracker option.

Subroutine STORE

STORE is a Fortran subroutine that stores displayed data from the CRT onto a specified disk for later use. The subroutine
is called when the store option is selected from the screen by the light pen pick.

**Subroutine STRTF**

STRTF is an assembly language subroutine that creates or starts a new file in the RANDOM IO subroutine package.

**Subroutine TOT**

TOT is a Fortran subroutine that sets up the accumulation array in the WABCOM subroutine.

**IMAGE TWERP**

TWERP is a Fortran IMAGE subroutine that displays the fuselage geometry generated by the subroutine SKETCH.

**Subroutine WABCOM**

WABCOM is a Fortran subroutine that calculates the mass properties of a flight vehicle with several component parts. There are two options in this subroutine, one being the geometry option which allows the user to input corner-point geometry and have the program output a black box component which is then summed into the total calculation of other black boxes. The other option is the black box option which allows a user to change the values of the mass properties of a component and reflect those changes to the total mass properties of the flight vehicle.

**Subroutine WABLST**

WABLST is a Fortran subroutine that generates the menus for the mass property data that is displayed on the CRT.

**IMAGE WABMNU**

WABMNU is a Fortran image subroutine that displays the numeric mass property data onto the CRT.

**Subroutine WABSET**

WABSET is a Fortran subroutine that sets up data into mass property arrays that are used by the WABCOM subroutine.

**Subroutine XFORM**

XFORM is a Fortran subroutine that makes coordinate transformations for the geometry option in WABCOM.
Subroutine XPOSE

XPOSE is a Fortran subroutine that takes the transpose of a three by three (3X3) matrix.

Subroutine $NALTS

$NALTS is an Adage system subroutine that halts the display of the present image.

Subroutine $PPLCOPY

$PPLCOPY is an Adage system subroutine that makes a hard copy of the present image displayed on the CRT.

Subroutine $PTCH9

This is an Adage system subroutine that specifies the CRT to be an input/output device. This subroutine needs to be called only once at the start of a program.

Subroutine $STOPS

$STOPS is an Adage system subroutine that stops the execution of the program loaded and returns the control of the monitor system of the Adage.
SUBROUTINE AN6

COMMON/ PARA/ LFLG(10), MFLAG(4), SUM, LETTER, NO, NFLG, INO, NFLAG(12)

* JJJ, KKK, LLL, MMM, NNN, NAME(2,60), NME(2), IINUM, RRNUM

DIMENSION TSM(20), SN(20), ESUM(20), MASK(5)

DATA TSM/20*0./, SUM/0./, LFLG(6)/0./, LFLG(3)/1/

DATA SN/20*0./, ESUM/20*0./

DATA MASK/0B, 7700000000B, 7777000000B, 7777700000B, 7777777700B/

IF(LFLG(3),EQ.0) GO TO 89

100 CALL ANRST
C A1.1
C ONE DATA ENTRY PER LIGHT PEN DEPRESS

89 IF(LFLG(6).EQ.1) GO TO 89
C A1.2
C WHEN CLR IS PICKED BY LIGHT PEN
C
10 IF(LFLG(9).EQ.1) GO TO 100
C A1.3
C CONSTRUCTION OF ALPHA WORD
C
14 IF(MFLAG(1).EQ.1) GO TO 60
C A1.4
C WHEN ENT IS PICKED BY LIGHT PEN
C
17 IF(LFLG(8).EQ.1) GO TO 999
C A1.5
C DETERMINE IF NUMERICAL INPUT IS REAL OR INTEGER
C
24 IF(LFLG(7).EQ.22.56.20
25 20 IF(LFLG(5).EQ.1) GO TO 35
26 K=K+1
27 IF(LFLG(1).NE.1) GO TO 25
28 ICNT=ICNT+1
29 II=ICNT
30 L=K+II
31 IF(ICNT.EQ.1) GO TO 24
32 GO TO 30
33 24 LFLG(7)=0
34 GO TO 999
35 25 II=K-1
36 30 TSUM(K)=SUM
37 SN(K)=10.**II
38 LFLG(7)=0
39 GO TO 999
40 35 M=M+1
41 ESUM(M)=SUM
42 LFLG(7)=0
43 GO TO 999
44 999 EXP=INUM=0
45 RNUM=0.
46 IF(LFLG(5).NE.1) GO TO 43
47 DO 42 I=1,M
48 J=I-M
49 EXP=EXP+ESUM(I)*10.**J
50 42 CONTINUE
51 43 DO 50 I=1,K
52 J=L+I-1
53 IF(I.GT.L) J=I
54 IF(LFLG(1).EQ.1) GO TO 45
55 J=K+I-1
56 INUM=INUM+TSUM(I)*SN(J)
57 GO TO 50

26
67  45  RNUM=RNUM+TSUM(I)*SN(J)*10.**EXP
70  50  CONTINUE
71  C    AI.6
72  C  TEST FOR NEGATIVE INPUT NUMBER
73  C
74    IF(LFLG(1).EQ.1) GO TO 53
75    IF(LFLG(2).EQ.1) INUM=-INUM
76    INUM=INUM
77    GO TO 55
100   53  IF(LFLG(2).EQ.1) RNUM=-RNUM
101    RRNUM=RNUM
102   55  IF(LFLG(8).EQ.0) RETURN
103    LFLG(3)=1
104   56  RETURN
105   60  IF(LFLG(9).NE.1) GO TO 61
106    LFLG(3)=1
107    RETURN
110   61  IF(ICNT.GE.10) RETURN
111    IF((LFLG(7).EQ.0).XOR.(LFLG(2).EQ.0)) GO TO 62
112    RETURN
113   62  ICNT=ICNT+1
114    KCNT=ICNT
115    J=1
116    IF(ICNT.LE.5) GO TO 63
117    J=2
120   63  NDX=6*(KCNT-1)
121   64  NME(J)=(NME(J).AND.MASK(KCNT)).OR.(.NOT.MASK(KCNT)).AND. (LETTER.R.NDX))
122    LFLG(7)=0
123    LFLG(2)=0
126    RETURN
SUBROUTINE ANRST
C A2.0
C RESET GRID INPUT VARIABLES
C
K=L=M=II=ICNT=IINUM=0
RRNUM=0.
NME(1)=NME(2)='
DO 60 I=1,10
LFLG(I)=0
60 CONTINUE
DO 70 I=1,20
ESUM(I)=0.
TSUM(I)=0.
70 CONTINUE
RETURN
END
SUBROUTINE AN7

CONSTRUCTION OF ALPHA/NUMERIC INPUT FOR WABCOM

COMMON/PARA/LFLGCIO),MFLAG(4),SUM,LETTER,NO,NFLAG,INO,NFLAG(12)
* ,JII, KKK,LLL,MMM,NNN,NAM(2,60),NME(2),IINUM,RRNUM
COMMON/LTR/IFLAG(40)
DIMENSION TSUM(20),SN(20),ESUM(20),MASK(5)
LOGICAL IFLAG
DATA TSUM/20*0./,SUM/0./,LFLG(6)/0./,LFLG(3)/1/
DATA SN/20*0./,ESUM/20*0./,JCNT/0/
DATA MASK/0B,7700000000B,7777000000B,7777770000B,7777777700B/
JCNT=JCNT+1
IF(JCNT.EQ.1) GO TO 100
IF(LFLAG(3).EQ.0) GO TO 89
100 CALL ANRST
C ONE DATA ENTRY PER LIGHT PEN PICK

99 IF(LFLG(6).EQ.1) GO TO 99

C WHEN CLR PICKED BY LIGHT PEN

C IF(LFLG(9).EQ.1) GO TO 100

C CONSTRUCTION OF ALPHA WORD

C IF(IFLAG(31)) GO TO 60

C WHEN ENT IS PICKED BY LIGHT PEN

C IF(LFLG(8).EQ.1) GO TO 999

C DETERMINE IF NUMERICAL INPUT IS REAL OR INTEGER

C IF(LFLG(7)) 22,56,20

20 IF(LFLG(5).EQ.1) GO TO 35

K=K+1

IF(LFLG(1).NE.1) GO TO 25

22 ICNT=ICNT+1

II=I-ICNT

L=K+II

50 999 EXP=INUM=0

RNUM=0.

IF(LFLG(5).NE.1) GO TO 43

DO 42 I=1,M

J=M-I

EXP=EXP+ESUM(I)*10,**(J)

CONTINUE

42 CONTINUE

IF(LFLG(4).EQ.1) EXP=-EXP

43 DO 50 I=1,K

J=L+I-1

IF(I.GT.L) J=I

IF(LFLG(1).EQ.1) GO TO 45

J=K+I-1

INUM=INUM+TSUM(I)*SN(J)

GO TO 50
45 RNUM=RNUM+TUM(I)*SN(J)+10.**EXP
50 CONTINUE
C  N1.6
C  TEST FOR NEGATIVE INPUT NUMBER
C
71 IF(LFLG(1).EQ.1) GO TO 53
72 IF(LFLG(2).EQ.1) INUM=-INUM
73 INUM=INUM
74 GO TO 55
75 IF(LFLG(2).EQ.1) RNUM=-RNUM
76 RRNUM=RNUM
77 IF(LFLG(8).EQ.0) RETURN
78 LFLG(3)=1
79 RETURN
100 IF(LFLG(2).EQ.1) RNUM=-RNUM
101 RRNUM=RNUM
102 IF(LFLG(8).EQ.0) RETURN
103 LFLG(3)=1
104 RETURN
105 IF(LFLG(8).NE.1) GO TO 61
106 LFLG(3)=1
107 RETURN
110 IF(ICNT.GE.10) RETURN
111 IF((LFLG(7).EQ.0).XOR.(LFLG(2).EQ.0)) GO TO 62
112 RETURN
113 ICNT=ICNT+1
114 KCNT=ICNT
115 J=1
116 IF(ICNT.LE.5) GO TO 63
117 J=2
118 KCNT=ICNT-5
119 NDX=6*(KCNT-1)
120 NME(J)=(NME(J).AND.MASK(KCNT)).OR.(.NOT.MASK(KCNT)).AND.
121 * (LETTER.R.NDX)
122 LFLG(7)=0
123 LFLG(2)=0
124 RETURN
SUBROUTINE ANRST

C    N1.7
C    RESET GRID INPUT VARIABLES
C
K=L=M=II=ICNT=IINUM=0
RRNUM=0.
NME(1)=NME(2)=''
DO 60 I=1,10
  LFLG(I)=0
60 CONTINUE
DO 70 I=1,20
  ESUM(I)=0.
  TSUM(I)=0.
70 CONTINUE
RETURN
END
IMAGE GRIDW

C
C
C

COMMON/PARA/ LFLG(10), MFLAG(4), SUM, LETTER, NO, NFLAG,INO, NFLAG(12)
* , JJJ.KKK.LLL.MMM.NNN.NAME(2,60), NHE(2), IINUM, RRNUM
COMMON/GRID/ BOX(50), TEST(200), TEST3(50), OPDPL(100), BOX(20)
INTEGER ALPHA(27),$PWD4
DATA ALPHA(1HA,1HG,1HM,1HS,1HB,1HH,1HN,1HT,1HC,1HI,1HO,1HU, 
* 1HD,1HJ,1HP,1HE,1HK,1HV,1HY,1HF,1HL,1HR,1HX,1HZ,1H/ 
DATA LFLG(10)/0/
LDX(1.F)
LDY(-1.F)
LSCL(1.4F)
TABLE2D(BOX)
CHAR(TESt3)
IF(LFLG(6).EQ.0) GO TO 200
IF(($LPNSW.AND.200B).NE.0) GO TO 200
LFLG(6)=0
200 CLEARPEN(1)
* $PUD4=0
26 PENON(1)

MENU("$80X00619Y0070A9.9A9X00660Y0070B9.9A9
* $80X00739Y0070M9.9A9X00773Y0070S9.9A9
* $80X00619Y0074B9.9A9X00660Y0074H9.9A9
* $80X00739Y0074N9.9A9X00779Y0074T9.9A9
* $80X00618Y0100C9.9A9X00668Y0100I9.9A9
* $80X00739Y010009.9A9X00779Y0100U9.9A9
* $80X00618Y0106D9.9A9X00660Y0106J9.9A9
* $80X00739Y0106P9.9A9X00779Y0106V9.9A9
* $80X00618Y0112E9.9A9X00660Y0112K9.9A9
* $80X00739Y0112B9.9A9X00779Y0112W9.9A9
* $80X01049Y0112Y9.9A9X00618Y0120F9.9A9
* $80X00669Y0120L9.9A9X00739Y0120R9.9A9
* $80X00779Y0120K9.9A9X01049Y0120Z9.9A9
* $80X01116Y0120/8.9A9
* $80X01049Y007019.9A9X01116Y007029.9A9
* $80X01116Y007039.9A9X01046Y007449.9A9
* $80X01116Y007459.9A9X01116Y007469.9A9
* $80X01046Y010079.9A9X01116Y010089.9A9
* $80X01116Y010099.9A9X01046Y010609.9A9
* $80X01116Y0106*9.9A9S0X01116Y0106CLR9.9A9
* $9S0X01116Y0112-9.9A9S0X01116Y0112EX9.9A9
* $9S0X01116Y0120ENT9.0Z")
PENOFl(1)
IF($PENHIT(1)) LFLG(10)=1
GO TO (16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16, 
* 16,16,16,16,16,16,16,16,16,16,16, 
* 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
RETURN
1 IF(LFLG(6).EQ.1) RETURN
SUM=1.
LETTER=1HI
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
2 IF(LFLG(6).EQ.1) RETURN
SUM=2.
LETTER=1H2
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
3 IF(LFLG(6).EQ.1) RETURN
SUM=3.
LETTER=1H3
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
4 IF(LFLG(6).EQ.1) RETURN
SUM=4.
LETTER=1H4
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
5 IF(LFLG(6).EQ.1) RETURN
SUM=5.
LETTER=1H5
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
6 IF(LFLG(6).EQ.1) RETURN
SUM=6.
LETTER=1H6
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
7 IF(LFLG(6).EQ.1) RETURN
SUM=7.
LETTER=1H7
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
8 IF(LFLG(6).EQ.1) RETURN
SUM=8.
LETTER=1H8
LFLG(6)=1
LFLG(7)=1
$PWD4=0
RETURN
9 IF(LFLG(6).EQ.1) RETURN
SUM=9.
155 LETTER=I*H9
156 LFLG(6)=1
157 LFLG(7)=1
160 $PWD4=0
161 RETURN
162 10 IF(LFLG(6).EQ.1) RETURN
163 SUM=0.
164 LETTER=I*H0
165 LFLG(6)=1
166 LFLG(7)=1
167 $PWD4=0
170 RETURN
171 11 IF(LFLG(6).EQ.1) RETURN
172 LFLG(1)=1
173 LFLG(6)=1
174 LFLG(7)=-1
175 LETTER=I*H*
176 $PWD4=0
177 RETURN
200 12 IF(LFLG(6).EQ.1) RETURN
201 LFLG(6)=1
202 LFLG(9)=1
203 $PWD4=0
204 RETURN
205 13 IF(LFLG(6).EQ.1) RETURN
206 IF(LFLG(5).NE.1) GO TO 18
207 LFLG(6)=1
208 LFLG(4)=1
209 LFLG(7)=0
212 $PWD4=0
213 RETURN
214 18 LFLG(2)=1
215 LFLG(7)=0
216 LFLG(6)=1
217 LETTER=I*H-
220 $PWD4=0
221 RETURN
222 14 IF(LFLG(6).EQ.1) RETURN
223 LFLG(6)=1
224 LFLG(5)=1
225 LFLG(7)=0
226 $PWD4=0
227 RETURN
230 15 IF(LFLG(6).EQ.1) RETURN
231 LFLG(6)=1
232 LFLG(8)=1
233 $PWD4=0
234 RETURN
235 16 IF(LFLG(6).EQ.1) RETURN
236 LFLG(6)=1
237 LFLG(7)=1
240 LETTER=ALPHA($PWD4)
241 $PWD4=0
242 RETURN
SUBROUTINE BLDNAM(NAM)

C FORMATS FILE INPUT NAME FOR FILE ACCESSING SUBROUTINES
C
IMPLICIT INTEGER (A-Z)
DIMENSION NAM(3)

DATA MASK/77B/

IF(NAM(2).EQ.1H) NAM(2)=NAM(1)
IF(NAM(1).EQ.NAM(2)) RETURN
NAM(3)=NAM(2)
NAM(2)=NAM(1)
DO 10 I=9,0,-1
WD=I/5+2
KNT=(6*I)/30
ISHFT=24-KNT
TM=MASK.L.ISHFT
POSC=NAM(WD).AND.TM
CHAR=(POSC.L.KNT).OR.1H
IF((CHAR.NE.1H).AND.((CHAR.NE.1H) .OR.1H)) GOTO 20
NAM(WD)=NAM(WD).AND.(-TM)
10 CONTINUE
20 CONTINUE
NAM(1)=(5*NAM+1).AND.77777B
NAM(1)=NAM(1).OR.777750000B
RETURN
END
SUBROUTINE BLDNME(NAME)

C FORMATS FILE INPUT NAME FOR FILE ACCESSING SUBROUTINES

C

DIMENSION L(10),NAME(3)
NINE=9
READ(NINE,100) L
100 FORMAT(10R1)
DO 3 M=1,3
3 NAME(M)=0
J=2
NL=0
NBLNK=0
DO 50 I=1,10
5 IF(L(I),EQ.408) GO TO 15
10 IF(NBLNK,NE.0) GO TO 20
5 IDUM=L(I)
CALL IT
GO TO 50
15 NBLNK=NBLNK+1
GO TO 50
20 IDUM=40B
DO 21 L=1,NBLNK
21 CALL IT
NBLNK=0
GO TO 5
50 CONTINUE
IF(NBLNK.EQ.0) GO TO 60
IDUM=0
DO 70 L=1,NBLNK
70 CALL IT
NBLNK=0
IF(NAME(3).NE.0) GO TO 80
NAME(1)=NAME(2)
RETURN
80 LOC=NAME(2)
NAME(1)=('H.77777B')
O. LOC
RETURN
C
SUBROUTINE IT
NAME(J)=.K.NAME(J)
NAME(J)=NAME(J).O.IDUM
NL=NL+1
IF(NL.GE.5) J=3
RETURN
END
SUBROUTINE CGMOVE

BLACK BOX TRACKER CG TRANSLATOR

COMMON/MOVIT/NAME(2),XCG,YCG,ZCG,NREAD
COMMON/STORE/CAMB,ALEN,RADZ(30),Rady(30),RADX(30),CAMX(30),
* CAMZ(30),FACTOR,IBLOCK,FILLER(56)
LOGICAL FLAG,ADTF1-,ADTF2,TOP,NREAD
LOGICAL FPEN,FPENP,NDSTP
FRACTION RADX,RADZ,SX,SY,RADTX,RADTY,POSTX,POSTY,POSTZ
FRACTION RX,RY,RY,BIAS,FDUM(30)
DIMENSION NAMDUM(3),IDUM(30),NAMES(50),NOTES(50),NOTES1(50),NOTES2(50)
DIMENSION TVIEW(100),SVIEW(100),DUMPIT(210)
EQUIVALENCE(DUMPIT,CAMB)
EQUIVALENCE(NPTS,RADZ(1)),(NPTSC,CAMZ(1)),(NPTS,RADY(1))

$NDSTP=.TRUE.

DINGER DATA

DATA2D(DINGER)
ZSET(1,F)
MOVE(1,F,1,F)
DRAW(1,F,-1,F)
DRAW(-1,F,-1,F)
DRAW(-1,F,1,F)
DRAW(1,F,1,F)
ENDLIST

ENDDATA

IF(NDSTP) GO TO 1
IF(NREAD) GO TO 2000

INITIALIZE CG POSITION

POSTX=-RTOF(XCG/FACTOR+.8)
POSTY=RTOF(YCG/FACTOR)
POSTZ=RTOF(ZCG/FACTOR)

INITIALIZE LABELS

LABEL(NOTES)
WRITE(16,40) NAME
FORMAT("$@AX@030@Y@014COMPONENT@X@074XCG@X115YCG@X135ZCG@X030@Y
*020",2A5,"@."),
ENDLIST
LABEL(NOTES1)
WRITE(16,41)XCG,ZCG
FORMAT("$@AX@030@Y@020",F7.1,"@X@125",F7.1,"@.")
ENDLIST
67  LABEL(NOTES2)
70  WRITE(16,42)YCG
71  42  FORMAT("988X91056Y9020",F7.1,2.")
72  ENDLIST
73  86  DISPLAY TWERP
74  1000  IF(DEPRESS(1)) GO TO 6000
75  C C1.4
76  C  WHERE IS DINGER?
77  C
100  RX=$RADTX
101  RY=$RADTY
102  TOP=.FALSE.
103  IF(RADTY.GT.0.F) TOP=.TRUE.
104  XPOS=FTOR(RX)+.8
105  BIAS=.6F
106  IF(TOP) BIAS=-.4F
107  YPOS=FTOR(RY+BIAS)
110  C C1.5
111  C DINGER POSITION IN ENGINEERING UNITS
112  C
113  ENGX=-XPOS*FACTOR
114  ENGZ=YPOS*FACTOR
115  DIMENSION POSN(10)
116  C C1.6
117  C DINGER POSITION ON SCREEN
120  C
121  LABEL(POSN)
122  IF(TOP) GO TO 36
123  WRITE(16,89) ENGX,ENGZ
124  89  FORMAT("99868X9150X",F6.0," Y=",F6.0)
125  GO TO 37
126  WRITE(16,91) ENGX,ENGZ
127  91  FORMAT("99868X9150X",F6.0," Z=",F6.0)
130  37  CONTINUE
131  ENDLIST
132  IF(.N.FLAG) GO TO 1000
133  NDSTP=.TRUE.
134  200  IF(NDSTP) GO TO 200
135  C C1.7
136  C MOVE CG TO NEW POSITION
137  C
140  IF(TOP) GO TO 210
141  POSTY=SY+.6F
142  YCG=FTOR(POSTY)*FACTOR
143  LABEL(NOTES2)
144  WRITE(16,42)YCG
145  ENDLIST
146  FLAG=.FALSE.
147  GO TO 86
150  210  POSTX=SX
151  POSTZ=SY-.4F
152  XCG=-(FTOR(POSTX)+.8)*FACTOR
153  ZCG=FTOR(POSTZ)*FACTOR
154  LABEL(NOTES1)
155   WRITE(16,41)XCG,ZCG
156   ENDLIST
157   FLAG=.FALSE.
160   GO TO 86
161   C
162   6000 CONTINUE
163   RETURN
1 2000 CALL $PTCH9
2 C C18
3 C GET FUSELAGE SHAPE FROM DISK
4 C
5 NINE=9
6 WRITE(NINE,2001)
7 2001 FORMAT(' ENTER PACK/VOL NO OF INPUT...
8 READ(NINE,0) LPV
9 LLPV=IDCT(LLPV)
10 2015 WRITE(NINE,2002)
11 2002 FORMAT(' ENTER NAME OF INPUT FILE...
12 CALL BLDNME(NAMDUM)
13 $NEWID=50000000008
14 $DFILETYPE=178
15 IFILAD=1
16 CALL RDSDKOK(LLPV,NAMDUM,208,IFILAD,IFLG)
17 IF(IFLG.NE.0) GO TO 2010
18 WRITE(10,2003)NAMDUM(2),FIAMDUM(2),LLPV
19 2003 FORMAT(' CAN NOT LOCATE ',A5,' ON PACK/VOL ',0)
20 GO TO 2010
21 2010 CALL RNDRD(DUMPIT,0,208,IFILAD,IOFF)
22 CALL DSKER
23 NEWID=70000000008
24 $DFILETYPE=78
25 CALL SKETCH
26 NREAD=.FALSE.
27 GO TO 72
28 C
29 C
30 C
31 C
32 C
33 C
34 C
SUBROUTINE SKETCH

CREATE FUSELAGE OUTLINE

ISS=NPTS+1
DO 2 IS=2,ISS
FDUM(IS)=-RADZ(IS)
2    IF(CAMB) FDUM(IS)=FDUM(IS)+CAMZ(IS)+CAMZC(IS)
LIST2D(SVIEW)
ZSET(0,F)
DO 3 IS=2,ISS
IF(IS.GT.2) GO TO 3
MOVE(RADX(IS),RADZ(IS))
DRAW(RADX(IS),RADZ(IS))
DO 4 IS=2,ISS
IF(IS.GT.2) GO TO 4
MOVE(RADX(IS),FDUM(IS))
DRAW(RADX(IS),FDUM(IS))
ENDLIST
LIST2D(TVIEW)
ZSET(0,F)
DO 5 IS=2,ISS
IF(IS.GT.2) GO TO 5
MOVE(RADX(IS),Rady(IS))
DRAW(RADX(IS),Rady(IS))
DO 6 IS=2,ISS
IF(IS.GT.2) GO TO 6
MOVE(RADX(IS),-Rady(IS))
DRAW(RADX(IS),-Rady(IS))
ENDLIST
RETURN
C

C:                                                          

C C2.0

C DISPLAY FUSELAGE

READ(3)

C CHECK DDT PEN FOR TIP SWITCH

FPEN=.FALSE.

IF(.N.*ADTF1 .A. *ADTF2) FPEN=.TRUE.

IF(FPEN.A..N.FPENP) FLAG=.TRUE.

IF(.N. FPEN) GO TO 5

C C2.2

SAVE DDT PEN POSITION

SX=$RADTX

SY=$RADTY

FPENP=FPEN

C C2.3

GRID DATA

DATA2D(GRD)

ZSET(1.F)

LINE(-.8F,0.F,.82F,0.F)

LINE(-.8F,0.F,-.8F,.41F)

LINE(-.8F,0.F,.82F,0.F)

LINE(-.8F,0.F,-.8F,.41F)

ENDLIST

ENDDATA

LDY(.4F)

C C2.4

SHOW SIDEVIEW & GRID

TABLE2D(SVIEW)

TABLE2D(GRD)

POSCHAR(-.82F,.42F,0.F,"S9B-Z")

POSCHAR(.83F,-.01F,0.F,"S9B-X")

POSCHAR(-.2F,.45F,0.F,"SBSIDEVIEW")

C C2.5

POSITION & DISPLAY SIDEVIEW CG

DY(POSTZ)

DX(POSTX)

LSCL(1.F)

TABLE2D(DINGER)

C C2.6

SHOW TOPVIEW & GRID

LSCL(1.F)

LDY(-.6F)

LDX(0.F)
67   TABLE2D(TVIEW)
70   TABLE2D(GRD)
71   POSCHAR(-.8F,.42F,0,F,"@S@BY")
72   POSCHAR(.82F,-.005F,0,F,"@S@S-X")
73   POSCHAR(-.2F,.45F,0,F,"@BPLANVIEW")
74   C  C2.7
75   C  POSITION & DISPLAY TOPVIEW CG
76   C
77   DX(POSTX)
100  DY(POSTY)
101  LSCL(.1F)
102  TABLE2D(DINGER)
103  LSCL(1.F)
104  CENTER
105  C  C2.8
106  C  WRITE NOTES ON SCREEN
107  C
110  CHAR(POSN)
111  CHAR(NOTES1)
112  CHAR(NOTES2)
113  CHAR(NOTES)
114  IF($ADTF1) RETURN
115  C  C2.9
116  C  DISPLAY DINGER
117  C
120  LDX(RX)
121  LDY(RY)
122  LSCL(.01F)
423  TABLE2D(DINGER)
124  RETURN
125  END
SUBROUTINE CROSS(A,B,C,CM)

UNITIZED MATRIX CROSS PRODUCT

C = A X B

DIMENSION A(3),B(3),C(3)

C(1)=A(2)*B(3)-A(3)*B(2)
C(2)=A(3)*B(1)-A(1)*B(3)
C(3)=A(1)*B(2)-A(2)*B(1)

CM=SQRT(C(1)*C(1)+C(2)*C(2)+C(3)*C(3))

RETURN

END
SUBROUTINE DEC FMT (VALUE,NFIELD,FMT)
C BUILD FORMAT FOR DECIMAL NUMBERS
C
INTEGER FMT,CVT(2),POINT
DATA (CVT(I),I=1,2) /1HF,1HE/
DATA POINT/1H./
C DETERMINE THE EXPONENT (NPWR
NPWR=0
ABVAL=ABS<(1.+10.**(3-NFIELD))*VALUE)
IF (ABVAL.LT.1.) GO TO 550
DO 530 I=1,99
ABVAL=ABVAL/10.
IF (ABVAL.LT.1.) GO TO 540
530 CONTINUE
540 CONTINUE
NPWR=I
GO TO 600
550 CONTINUE
DO 570 I=1,99
ABVAL=ABVAL*10.
IF (ABVAL.GT..1) GO TO 580
570 CONTINUE
NPWR=-I
SELECT THE FORMAT
600 CONTINUE
NSP=1
IF (VALUE.LT.0) NSP=2
IF (NPWR.GE.0) GO TO 620
IF (NPWR.LT.-4) GO TO 640
NDEC=NFIELD-NSP
IFMT=1
GO TO 690
620 CONTINUE
NDEC=NFIELD-NPWR-NSP
IF (NDEC.LE.0) GO TO 640
IFMT=1
GO TO 690
640 CONTINUE
NDEC=NFIELD-NSP-5
IFMT=2
690 CONTINUE
IF (NDEC.Lt.0) NDEC=0
BUILD THE FORMAT
ENCODE(5,1000,FMT) CVT(IFMT),NFIELD,POINT,NDEC
1000 FORMAT (A1,I2,A1,I1)
RETURN
END
LFLG(7)=1
PWN4=0
RETURN

2 IF(LFLG(6).EQ.1) RETURN
SUM=2.
LETTER=IH2
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

3 IF(LFLG(6).EQ.1) RETURN
SUM=3.
LETTER=IH3
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

4 IF(LFLG(6).EQ.1) RETURN
SUM=4.
LETTER=IH4
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

5 IF(LFLG(6).EQ.1) RETURN
SUM=5.
LETTER=IH5
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

6 IF(LFLG(6).EQ.1) RETURN
SUM=6.
LETTER=IH6
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

7 IF(LFLG(6).EQ.1) RETURN
SUM=7.
LETTER=IH7
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

8 IF(LFLG(6).EQ.1) RETURN
SUM=8.
LETTER=IH8
LFLG(6)=1
LFLG(7)=1
PWN4=0
RETURN

9 IF(LFLG(6).EQ.1) RETURN
SUM=9.
155  LETTER=IH9
156   LFLG(6)=1
157   LFLG(7)=1
160   $PWD4=0
161   RETURN
162 10 IF(LFLG(6).EQ.1) RETURN
163   SUM=0.
164   LETTER=IH0
165   LFLG(6)=1
166   LFLG(7)=1
167   $PWD4=0
170   RETURN
171 11 IF(LFLG(6).EQ.1) RETURN
172   LFLG(1)=1
173   LFLG(6)=1
174   LFLG(7)=-1
175   LETTER=IH-
176   $PWD4=0
177   RETURN
200 12 IF(LFLG(6).EQ.1) RETURN
201   LFLG(6)=1
202   LFLG(9)=1
203   $PWD4=0
204   RETURN
205 13 IF(LFLG(6).EQ.1) RETURN
206   IF(LFLG(5).NE.1) GO TO 18
207   LFLG(6)=1
210   LFLG(4)=1
211   LFLG(7)=0
212   $PWD4=0
213   RETURN
214 18 LFLG(2)=1
215   LFLG(7)=0
216   LFLG(6)=1
217   LETTER=IH-
220   $PWD4=0
221   RETURN
222 14 IF(LFLG(6).EQ.1) RETURN
223   LFLG(6)=1
224   LFLG(5)=1
225   LFLG(7)=0
226   $PWD4=0
227   RETURN
230 15 IF(LFLG(6).EQ.1) RETURN
231   LFLG(6)=1
232   LFLG(8)=1
233   $PWD4=0
234   RETURN
235 16 IF(LFLG(6).EQ.1) RETURN
236   LFLG(6)=1
237   LFLG(7)=1
240   LETTER=ALPHA($PWD4)
241   $PWD4=0
242   RETURN
FUNCTION IOCT(NN)

C U1.0
C COMPUTES OCTAL EQUIVALENCE FROM DECIMAL INPUT
C
ITMP=0
ITMP=\left(\frac{\text{NN}}{1000}\right) \times 512 + \text{ITMP}
NRMD=\text{NN} \div 1000
NTMP=\left(\frac{\text{NRMD}}{100}\right) \times 64 + \text{ITMP}
\text{ITMP}=\text{NRMD} \div 100
\text{JTMP}=\left(\frac{\text{ITMP}}{10}\right) \times 8 + \text{NTMP}
\text{IOCT}=\left(\frac{\text{ITMP}}{10}\right) + \text{JTMP}
RETURN
END
OPTIONS   VER 1   REV B   16 JUN 76

1
2 C 01.0
3 C DISPLAY PROGRAM OPTIONS
4 C
5 COMMON/PARA/ LFLG(10), MFLAG(4), SUM, LETTER, NO, NFLG, INO, NFLAG(12),
6 * JJJ, KKK, LLL, MMM, NNN, NAME(2,60), NME(2), IINUM, RNUM
7 COMMON/OPT/ NHIT, OPDPL
8 INTEGER *PWD4
9 DATA NHIT/0/
10 DATA2D (BOX)
11 ZSET(0.F)
12 MOVE(.4F,.4F)
13 DRAW(.4F,-1.F)
14 DRAW(-.4F,-1.F)
15 DRAW(-.4F,.4F)
16 DRAW(.4F,.4F)
17 ENDLIST
18 ENDDATA
19 LDX(1.F)
20 LDY(1.F)
21 TABLE2D (BOX)
22 CHAR("880X90710Y0064OPTIONS")
23 CHAR(OPDPL)
24 IF(NHIT.EQ.0) GO TO 100
25 IF((LPNSW.AND.200B).NE.0) GO TO 100
26 NHIT=0
27 100 CLEARPEN(1)
28 *PWD4=0
29 PENCN(1)
30 MENU("880X90619Y0072PACK/VOLS.8A")
31 "880X90619Y0076SELECT FILES.8A"
32 "880X90619Y0102LIST VOLS.8A"
33 "880X90619Y0106BLACK BOX.8A"
34 "880X90619Y0112GEOMETRY.8A"
35 "880X90619Y0116DISPLAY.8A"
36 "880X90619Y0122APPEND DISPLAY.8A"
37 "880X90619Y0126DELETE COMPONENT.8A"
38 "880X90619Y0132STORE.8A"
39 "880X90619Y0136INPUT NEW FILE.8A"
40 "880X90619Y0142BLACK BOX TRACKER.8A"
41 "880X90619Y0146PRINT.8A"
42 PENOFF(1)
43 IF((LF.GE(10)).OR.(MFLAG(1).EQ.1)).OR.(MFLAG(2).EQ.1))GOTO 200
44 IF(*PWD4.EQ.0) GO TO 200
45 IF(NHIT.EQ.1) GO TO 200
46 NFLAG(*PWD4)=1
47 MFLAG(3)=1
48 *PWD4=0
49 200 CALLSIMG GRID
50 RETURN
51 END
SUBROUTINE LOCFLE (IPV,NAMER,IFILNO)
COMMON/PARA/ LFLAG(10),MFLAG(4),SUM,LETTER,N0,NFLG,INO,NFLAG(12)
   *,JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),INUM,RRNUM
COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
   * VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
   * VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
REAL VALUE1,VALUE2,VALUE3,VALUE4,VALUES,
   * VALUE5,VALUE6,VALUE7,VALUE8,VALUE9,VALUE10,
   * VALUE11,VALUE12,VALUE13
INTEGER NAME,NAMER(3)
CALL BLDNAM(NAMER)
CALL DSVOL(IPV,IVFLAG,NFREES)
CALL DSKER
C WRITE(10,100) IVFLAG,NFREES
C 100 FORMAT(1X,5013)
CALL DSFIL(IPV,NAMER,IFFLAG,NMFL,IFILAD,IFILNO)
CALL DSKER
C WRITE(10,100) IFFLAG,NMFL,IFILAD,IFILNO
RETURN
END
SUBROUTINE INPUT(IPV)
C
C INPUT NEW DATA FILE
C
COMMON/STR/, NTOT, VALUE1(60), VALUE2(60), VALUE3(60), VALUE4(60),
* VALUE5(60), VALUE6(60), VALUE7(60), VALUE8(60), VALUE9(60),
* VALUE10(60), VALUE11(60), VALUE12(60), VALUE13(60)
DIMENSION IBUF(1000), DUMN(2)
REAL VALUE1, VALUE2, VALUE3, VALUE4, VALUE5,
* VALUE6, VALUE7, VALUE8, VALUE9, VALUE10,
* VALUE11, VALUE12, VALUE13
INTEGER NAME(2, 60), NAMER(3)
DATA DUMN(1)/5H 	 /, DUMN(2)/5H 	 /, DUMX/0./, DUMY/0./, DUMZ/0./
DATA NINE/9/
DATA FMT2/15151515156/, FMT3/77777777776/
DATA FMT4/77777000008/
NTOT=0
CALL PTCH9
WRITE(NINE, 3)
3 FORMAT(//,'	 ENTER NAME OF NEW FILE...')
READ(NINE, 5) NAMER
5 FORMAT(2A5)
CALL BLDNAM(NAMER)
10 WRITE(NINE,1776)
1776 FORMAT(/,'	 ENTER 13 VALUES...')
READ(NINE, 100) NAME(1, NTOT), NAME(2, NTOT)
100 FORMAT(2A5)
IF(NAME(1, NTOT) . EQ. "DONE") GO TO 25
WRITE(NINE, 1976)
1976 FORMAT(/,'	 ENTER 13 VALUES...')
READ(NINE, 0) VALUE1(NTOT), VALUE2(NTOT), VALUE3(NTOT)
* VALUE4(NTOT), VALUE5(NTOT), VALUE6(NTOT), VALUE7(NTOT)
* VALUE8(NTOT), VALUE9(NTOT), VALUE10(NTOT), VALUE11(NTOT)
* VALUE12(NTOT), VALUE13(NTOT)
GO TO 10
25 NTOT=NTOT-1
ENCODE(5,1000, IBUF(1)) NTOT, FMT2
50 100 FORMAT(A3, A2)
J=2
DO 5000 I=1, NTOT
ENCODE(75,2000, IBUF(J)) NAME(1,I), NAME(2,I), VALUE1(I), VALUE2(I),
* VALUE3(I), VALUE4(I), VALUE5(I), VALUE6(I), FMT2
5000 FORMAT(2A5, 6F10.1, A5)
ENCODE(75,3000, IBUF(J+15)) VALUE7(I), VALUE8(I), VALUE9(I),
* VALUE10(I), VALUE11(I), VALUE12(I), VALUE13(I), FMT2
6000 FORMAT(7F10.1, A5)
ENCODE(45,7000, IBUF(J+30)) DUMN(1), DUMN(2), DUMX, DUMY, DUMZ, FMT2
7000 FORMAT(2A5, 3F10.1, A5)
J=J+39
5000 CONTINUE
ENCODE(5, 4000, IBUF(J)) FMT4
4000 FORMAT(A5)
ENCODE(5, 6000, IBUF(J+1)) FMT3

6000 FORMAT(A5)

LENGTH=J

CALL STRTF(IPV, NAMER)

CALL APNDF(IBUF(1), LENGTH)

CALL APNDF(IBUF(J+1), 1)

CALL CLOSF

RETURN

END
SUBROUTINE LOAD(A,T,N)
DIMENSION T(I),A(I)
DO 1 I=1,N
   T(I)=A(I)
1 CONTINUE
RETURN
END
PROGRAM MAIN

C MAIN CONTROL FOR IMP

COMMON/ PARA/ LFLG(10),MFLAG(4),SUM,LETTER,NO,NFLAG,INO,NFLAG(12)
* JJJ,KKK,LLL,MMM,NNN,NAMES,60,NAMES(2),INUM,RRNUM
COMMON/ GRD/ BBOX(50),TEST,TEST3
COMMON/ CNT/ IPASS,NUP,NAMES(3),IPV,IENRT,IEND,IT,LINES
COMMON/ NE S/ NMBB(2)
COMMON/ MOV/ NAMDT(2),XCG,YCG,ZCG, IDTFLG
COMMON/ OPT/ NHIT,OPDPL
COMMON/ WB/ NMB(2),W(15),TEMP(18),WRITE,INN,ITN(2),TIXX,15,NUM
* IPRT,PWT(3),DCG(65,3),IGNME(3,6)
COMMON/ STR/ NTO,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
* VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
* VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
* REAL VALUE1,VALUE2,VALUE3,VALUE4,VALUE5,VALUE6,VALUE7,VALUE8,
* VALUE9,VALUE10,VALUE11,VALUE12,VALUE13
DIMENSION TEST(200),V(60,13),OPDPL(20),NMRDPL(2)
EQUIVALENCE (V(1,1),VALUE1(1))
INTEGER NAME,NAMES,NME,IPAR(5),NMDPL(2),IAPN(5)
LOGICAL IDTFLG
DATA INUM/0/,RRNUM/0/,NME/2/*,'MFLAG/4*0/,IDTFLG/.TRUE./
DATA IPAR/1H,4H WT,4HX-CG,4HY-CG,4HZ-CG,IDPV/110/,IPv/1108/
DATA IANP/SH NAME,SH WT,SH X-CG,SH Y-CG,SH Z-CG/
DATA NFLAG/12*0/,LFLG(10)/0/,LINES/20/,NMRDPL(1)/SHDATA/
DATA NMRDPL(2)/SH /* PWT/3*0.*/
DATA2D (BBOX)
ZSET(O.F)
MOVE(1.F,1.F)
DRAW(1.F,-1.F)
DRAW(-1.F,-1.F)
DRAW(-1.F,1.F)
DRAW(1.F,1.F)
MOVE(1.F,7142F)
DRAW(-1.F,7142F)
MOVE(-1.F,4285F)
DRAW(1.F,4285F)
MOVE(1.F,1428F)
DRAW(-1.F,1428F)
MOVE(-1.F,-1428F)
DRAW(1.F,-1428F)
MOVE(1.F,-4285F)
DRAW(-1.F,-4285F)
MOVE(-1.F,-7142F)
DRAW(1.F,-7142F)
MOVE(-1.F,7142F)
MOVE(-1.F,-1.F)
100  CALL RESET2
50  CALL RESET1
100  LABEL (TEST)
    ZSET(0.0F)
32  IF((MFLAG(1).EQ.0).AND.(MFLAG(2).EQ.0)) GO TO 130
33  WRITE(16,120) NMDSPL(1),NMDSPL(2),IPARA(INO)
120  FORMAT("GBG0600Y060",2A5,2X,A5)
130  IF(JJJ.NE.0) GO TO 170
36  IF(LFLG(1).EQ.1) GO TO 150
37  WRITE(16,140) IINUM
140  FORMAT("GBG000700Y0064",I10)
150  WRITE(16,160) RNUM
160  FORMAT("GBG000700Y0064",F10.5)
170  WRITE(16,180) NME(1),NME(2)
180  FORMAT("GBG000700Y0064",2A5)
190  ENDLIST
50  LABEL(OPDPL)
70  ZSET(0.0F)
80  WRITE(16,810) IDPV,NMRDPL(1),NMRDPL(2)
100  FORMAT("GBG0007077Y9072",I4,"9X0779Y076",2A5)
100  ENDLIST
55  M1.2
56  CALL $NHALT
1        CALL STOPS
2        CALL EXIT
3  200 IF(MFLAG(3).EQ.0) GO TO 195
4        CALL AN6
5 C     M1.3
6 C     PACK/ Vol option
7 C
8          IF(NFLAG(1).EQ.0) GO TO 500
9          MFLAG(2)=1
10         NO=1
11         NMDSPL(1)=SHENTER
12         NMDSPL(2)=4H IPV
13         IF(LFLG(3).NE.1) GO TO 100
14         IDPV=IINUM
15         IPV=IOCT(IINUM)
16         GO TO 40
17 C     M1.4
18 C     FILE NAME OPTION
19 C
20  500 IF(NFLAG(2).EQ.0) GO TO 510
21          MFLAG(1)=1
22         NO=1
23         NMDSPL(1)=SHENTER
24         NMDSPL(2)=5H FILE
25         JJJ=1
26          IF(LFLG(3).NE.1) GO TO 100
27         NAMER(1)=NME(1)
28         NAMER(2)=NME(2)
29         NMIDPL(1)=NME(1)
30         NMIDPL(2)=NME(2)
31         GO TO 40
32 C     M1.5
33 C     LIST VOLUME OPTION
34 C
35  510 IF(NFLAG(3).EQ.0) GO TO 520
36         NO=1
37         NMDSPL(1)=5HTBD
38         NMDSPL(2)=SHENTER
39          IF(LFLG(3).NE.1) GO TO 100
40         GO TO 40
41 C     M1.6
42 C     BLACK BOX OPTION
43 C
44  520 IF(NFLAG(4).EQ.0) GO TO 530
45          NFLAG(6)=O
46          IF((MFLAG(1).EQ.0).AND.(MFLAG(2).EQ.0)) GO TO 100
47 DO 525 I=1,2
48          NMB(I)=NAME(I,NO)
49 CONTINUE
50 DO 526 J=1,13
51          WT(J)=V(NO,J)
52          IF((J.LT.2).OR.(J.GT.4)) GO TO 526
53         I=I-1
54         PWR(I)=V(NO,J)
55 CONTINUE
67  526 CONTINUE
70  527 IF(LFLG(6).EQ.1) GO TO 527
71      CALL WABCOM
72      NAME(1,NO)=NMBB(1)
73      NAME(2,NO)=NMBB(2)
DO 528 J=1,13
V(NO,J)=WT(J)
IF(J.LT.2).OR.(J.GT.4) GO TO 528
I=J-1
DCG(NO,I)=DCG(NO,I)+(V(NO,J)-PWT(I))
528 CONTINUE
NFLAG(4)=0
NUP=1
NFLAG(6)=1
GO TO 50
C M1.7
C GEOMETRY OPTION
C
530 IF(NFLAG(5).EQ.0) GO TO 540
NFLAG(4)=NFLAG(6)=0
CALL WACOM
NTOT=NTOT+1
NAME(1,NTOT)=NMBB(1)
NAME(2,NTOT)=NMBB(2)
DO 535 J=1,13
V(NTOT,J)=WT(J)
535 CONTINUE.
DO 536 I=1,3
DCG(NTOT,I)=0,
536 CONTINUE
NFLAG(5)=0
NFLAG(6)=1
NUP=1
GO TO 50
C M1.8
C DISPLAY OPTION
C
540 IF(NFLAG(6).EQ.0) GO TO 550
IF(NUP.EQ.1) GO TO 543
IF(IPASS.NE.0) GO TO 544
NAME(1)=NRMDP(1)
NAME(2)=NRMDP(2)
CALL Lecute(I,NAME,IFILNO)
OPEN(20,IPV,IFILNO)
READ(20,541) NTOT
541 FORMAT(I3)
READ(20,542) (NAME(1,I),NAME(2,I),VALUE1(I),VALUE2(I),VALUE3(I),
VALUE4(I),VALUES(I),VALUE5(I),VALUE6(I),VALUE7(I),VALUES(I),VALUE8(I),
VALUE9(I),VALUES(I),VALUE10(I),VALUE11(I),VALUE12(I),VALUES(I),VALUE13(I),
IGNME(1,I),IGNME(2,I),DCG(I,J),J=1,3),I=1,NTOT)
542 FORMAT(2A5,6F10.1,/,2A5,3F10.1)
CLOSE(20)
543 CALL WASET
544 ITEMP=NTOT//LINES
IRMEDR=NTOT//LINES
IF(IRMEDR.NE.0) ITEMP=ITEMP+1
IF(.NOT.DEPRESS(32)) GO TO 546
IT=IT+1
IPASS=0
346 IF(IT.GE.ITEMP) IT=0
ISTRT=1+(IT*LINES)
IEND=LINES+(IT*LINES)
IF(IEND.GT.NTOT) IEND=NTOT
CALL WABLST
IF(<MFLAG<1).EQ.1).OR.(MFLAG(2).EQ.1)) GO TO 547
DO 545 I=1,12
IF(I.EQ.6) GO TO 545
IF(NFLAG(I).NE.1) GO TO 545
NFLAG(6)=0
GO TO 50
545 CONTINUE
547 NMDSPL(1)=NAME(1,NO)
NMDSPL(2)=NAME(2,NO)
IF(LFLG(3).EQ.0) GO TO 100
GO TO 50
M1.9
APPEND DISPLAY OPTION
C
550 IF(NFLAG(7).EQ.0) GO TO 560
IF(Ix.EQ.6) GO TO 559
MFLAG(2)=1
IF(Ix.NE.1) GO TO 552
MFLAG(1)=JJJ=1
552 NMDSPL(1)=SHENTER
NMDSPL(2)=IAPN(Ix)
IF(LFLG(3).NE.1) GO TO 100
555 NTOT=NTOT+1
NAME(1,NTOT)=NAME(1)
NAME(2,NTOT)=NAME(2)
GO TO 557
555 J=Ix-1
V(NTOT,J)=RRNUM
IF(J.EQ.1) GO TO 557
I=J-1
557 IX=IX+1
GO TO 50
559 DO 558 J=5,13
V(NTOT,J)=0.
558 CONTINUE
DO 556 I=1,2
55 CONTINUE
IGNME(I,NTOT)=5H
556 CONTINUE
IPASS=0
NUP=1
IX=1
NFLAG(7)=0
NFLAG(6)=1
GO TO 50

C M1.10
C DELETE COMPONENT OPTION
C
560 IF(NFLAG(8).EQ.0) GO TO 570
IF(NFLAG(1).EQ.0).AND.(NFLAG(2).EQ.0)) GO TO 100
NTOT=NTOT-1
DO 566 I=NO,NTOT
DO 564 J=1,2
NAME(J,I)=NAME(J,I+1)
IGNME(J,I)=IGNME(J,I+1)
564 CONTINUE
DO 565 K=1,13
V(I,K)=V(I+1,K)
IF(K.LT.2).OR.(K.GT.4)) GO TO 565
N=K-1
DCG(I,N)=DCG(I+1,N)
565 CONTINUE
566 CONTINUE
IPASS=0
NUP=1
NFLAG(8)=0
NFLAG(6)=1
GO TO 50

M1.11
C STORE: OPTION
C
570 IF(NFLAG(9).EQ.0) GO TO 580
NAMER(1)=NMRDPL(1)
NAMER(2)=NMRDPL(2)
.CALL STORE(IPV,NAMER)
NFLAG(9)=0
NFLAG(6)=1
GO TO 50
C M1.12
C INPUT NEW DATA FILE OPTION
C
580 IF(NFLAG(10).EQ.0) GO TO 590
CALL INPUT(IPV)
GO TO 30
C M1.13
C BLACK BOX TRACKER OPTION
C
590 IF(NFLAG(11).EQ.0) GO TO 595
IF((MFLAC(1).EQ.0).AND.(MFLAG(2).EQ.0)) GO TO 100
NAMDT(1)=NAME(1,NO)
NAMDT(2)=NAME(2,NO)
XCG=V(NO,2)
YCG=V(NO,3)
ZCG=V(NO,4)
DG 593 J=2,4
I=J-1
PWT(I)=V(NO,J)
67  593 CONTINUE
70    CALL CGMOVE
71    V(NO.2)=XCG
72    V(NO.3)=YCG
73    V(NO.4)=ZCG
DO 594 J=2,4
I=J-1
DCG(NO,I)=DCG(NO,I)+(V(NO,J)-PWT(I))
594 CONTINUE
NFLAG(11)=0
NUP=1
NFLAG(6)=1
GO TO 50
C M1.14
PRINT OPTION
C
595 IF(NFLAG(12).EQ.0) GO TO 100
CALL $PPLCOPY
NFLAG(12)=0
NFLAG(6)=1
GO TO 50
SUBROUTINE RESET1
C M2.0
C FIRST INITIALIZATION OF MAIN PROGRAM VARIABLES
C
MFLAG(1)=IINUM=IT=O
LLL=KKK=MMM=IWRITE=IPRINT=O
INO=1
NUMO=9
DO 600 I=1,3
PWT(I)=0.
RETURN
SUBROUTINE RESET2
C M3.0
C SECOND INITIALIZATION OF MAIN PROGRAM VARIABLES
C
MFLAG(3)=IINUM=IPASS=IT=O
IX=1
RRNUM=0.
NMDSPL(1)=NMDSPL(2)=5H
DO 1 I=1,12
NFLAG(I)=O
1 CONTINUE
RETURN
END
SUBROUTINE MXM3X3(A,B,C)
C C = AB ALL 3X3 MATRICES
DIMENSION A(9),B(9),C(9)
C(1)=A(1)*B(1)+A(4)*B(2)+A(7)*B(3)
C(2)=A(2)*B(1)+A(5)*B(2)+A(8)*B(3)
C(3)=A(3)*B(1)+A(6)*B(2)+A(9)*B(3)
C(4)=A(1)*B(4)+A(4)*B(5)+A(7)*B(6)
C(5)=A(2)*B(4)+A(5)*B(5)+A(8)*B(6)
C(6)=A(3)*B(4)+A(6)*B(5)+A(9)*B(6)
C(7)=A(1)*B(7)+A(4)*B(8)+A(7)*B(9)
C(8)=A(2)*B(7)+A(5)*B(8)+A(8)*B(9)
C(9)=A(3)*B(7)+A(6)*B(8)+A(9)*B(9)
RETURN
END
SUBROUTINE MXV(M, V, VV)

C THIS ROUTINE MULTIPLIES A 3X3 MATRIX TIMES A 3X1 MATRIX AND RETURNS
THE RESULTING 3X1 MATRIX.

C INPUTS:
M - 3X3 INPUT MATRIX
V - 3X1 INPUT MATRIX (VECTOR)

C OUTPUT:
VV = 3X1 OUTPUT. MAY BE STORED IN SAME LOCATION
AS INPUT VECTOR.

DIMENSION V(3), TMP(3), VV(3)
REAL M(9)

TMP(1) = M(1)*V(1) + M(4)*V(2) + M(7)*V(3)
TMP(2) = M(2)*V(1) + M(5)*V(2) + M(8)*V(3)
TMP(3) = M(3)*V(1) + M(6)*V(2) + M(9)*V(3)

DO 5 I=1,3
  VV(I) = TMP(I)
5 CONTINUE
RETURN
END
SUBROUTINE PAT(A,W,R,B)
C TRANSFORMS INERTIAS BY PARALLEL AXIS THEOREM
C
DIMENSION A(6),R(3),B(6)
B(1)=A(1)+W*(R(2)**2+R(3)**2)
B(2)=A(2)+W*(R(1)**2+R(3)**2)
B(3)=A(3)+W*(R(1)**2+R(2)**2)
B(4)=A(4)+W*R(2)*R(1)
B(5)=A(5)+W*R(1)*R(3)
B(6)=A(6)+W*R(3)*R(2)
RETURN
END
SUBROUTINE STORE(IPV,NAMER)
C  STORE DATA TO DISK
C
COMMON/Para/ LFLG(10),MFLAG(4),SUM,LETTER,NO,NFLG,INO,NFLAG(12)
* ,JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),INUM,RRNUM
COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
* VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
* VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
COMMON/WB/ NMB(2),WT(15),TEMP(18),IWRITE,IJ,ITNM(2),TIXX(15),NUMO
* ,IPRINT,PWT(3),DCG(60,3),IGNME(3,60)
DIMENSION IBUF(3000)
REAL VALUE1,VALUE2,VALUE3,VALUE4,VALUE5,VALUE6,VALUE7,VALUE8,
* VALUE9,VALUE10,VALUE11,VALUE12,VALUE13
INTEGER NAME,NAMER(3)
DATA FMT2/1515151515B/,FMT3/7777777777B/,FMT4/7777000000B/
CALL BLDNMAM(NAMER)
ENCODE(5,1000,IBUF(1)) NTOT,FMT2
1000 FORMAT(I3,A2)
J=2
DO 5000 I=1,NTOT
ENCODE(75,2000,IBUF(J)) NAME(I,1),NAME(I,2),VALUE1(I),VALUE2(I),
* VALUE3(I),VALUE4(I),VALUE5(I),VALUE6(I),FMT2
2000 FORMAT(2A5,6F10.1,A5)
ENCODE(75,3000,IBUF(J+15)) VALUE7(I),VALUE8(I),VALUE9(I),
* VALUE10(I),VALUE11(I),VALUE12(I),VALUE13(I),FMT2
3000 FORMAT(7F10.1,A5)
ENCODE(45,7000,IBUF(J+30)) IGNME(I,1),IGNME(I,2),(DCG(I,K),
* K=1,3),FMT2
7000 FORMAT(2A5,3F10.1,A5)
J=J+39
5000 CONTINUE
ENCODE(5,4000,IBUF(J))FMT4
4000 FORMAT(A5)
ENCODE(5,6000,IBUF(J+1))FMT3
45 FORMAT(A5)
LENGTH=J
CALL STRTF(IPV,NAMER)
CALL APNDF(IBUF(1),LENGTH)
CALL AFNDF(IBUF(J+1),1)
CALL CLOSF
RETURN
END
SUBROUTINE TOT(A,B,N)
DIMENSION A(I),B(I)
DO 1 I=1,N
A(I)=A(I)+B(I)
B(I)=0.
1 CONTINUE
RETURN
END
INCLUDE=DISK,TTY,PPL

SUBROUTINE WABCOM

COMMON/PARA/
LFLG(IO),MFLAG(4),SUM,LETTER,NO,NFLG,INO,NFLAG(12),
* JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),INUM,RRNUM
COMMON/WB/
NMB(2),WT(15),TEMP(18),IWRITE,IJ,ITNM(2),TIxx(15),NUMO
* ,IPRT,PUT(3),DCG(60,3),IGNMC(3,60)
COMMON/GRD/
BBX(50),TEST(200),TEST3(50),OPDPL(100),BOX(20)
COMMON/LTR/
IFLAG(40)
COMMON/NMES/
NMB(2)
COMMON/STR/
NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
* VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
* VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
DIMENSION LI(3),XIN(4),YN(4),ZIN(4),L2(3),D1(3),ZU(3)
DIMENSION YU(3),XU(3),TEMP(9),DC(9),PMIP(9),PMI(9)
DIMENSION DELX(3),XO(3),XBAR(3),XBRF(3),GIN(100)
DIMENSION SIXX(15),D2R(9),XN(30),BIN(150)
DIMENSION YA(30),ZC(30),XBC(30),YB(30),ZB(30)
DIMENSION FACT(10),NMB(6),IFILE(3),IFILEh(3)
EQUIVALENCE (IYY,WT(6)),(IZZ,WT(7)),(IXY,WT(8)),(IXZ,WT(9))
EQUIVALENCE (IY2,WT(10)),(S,WT(11)),(IXX,WT(3))
EQUIVALENCE (AREA,WT(13)),(VOL,WT(12)),(DELX,DELX(2))
* ,DELZ,DELX(3)
EQUIVALENCE (FACI,FACT(1)),(FACW,FACT(3))
EQUIVALENCE (FACG,FACT(4)),(PSI,FACT(8)),(THETA,FACT(9))
EQUIVALENCE (PHI,FACT(10))
REAL LI,LZ,IX,YY,IZZ,IXY,IXZ,IFY
LOGICAL IFLAG,SUB,BLKEN,RFILAG,RFLAG,BFLAG
INTEGER STAT,ROOT,STNAME,STAT,OVF
DATA DELX/3*0./PSI,PHI,THETA/3*0./RHO,H/2*1./
DATA FACI/1./FACW/1./FACG/1./FACL/12./,TEMP/18*0./
DATA IFLAG/40*.FALSE./,TIX,X/15 0./,IPRT/4T/O/, IBASE/0/
DATA NMB/10/4HCOMP/,STNAME/4HSUBT/,IWRITE/0/,IREFL/(0
DATA WT/15*0./,SUB./FALSE./,IFLP/0/
DATA NME/1108/,,NMB(2)/,,OVF/1108/,IPV/1108/
DATA NMB/5HENTER,5H VALU,5H ,5H ENTER,5H NAME,5H /
DATA IFILE/5HDATA ,5H ,5H /
C FORMAT STATEMENTS
C 1000 FORMAT(1X,2A5,6H WT = ,G13.7,6H XCG = ,G13.7,5X,
* SHYG = ,G13.7,5X,SHZCG = ,G13.7,6H IX = ,G13.7,5X,
* SHN = ,G13.7,5X,SHIZZ = ,G13.7,5X,SHXY = ,G13.7,
* SHIXX = ,G13.7,5X,SHIXZ = ,G13.7)
1011 FORMAT(6H ASF = ,G13.7,5X,SHVOL = ,G13.7,5X,SHAFR = ,G13.7)
1500 FORMAT(2(3F10.0,1))
1700 FORMAT(25)
1710 FORMAT(I5)
C NUMO=9
67       IF(IJ.GT.1) GO TO 6000
70       NTIME=0
71  6000 IF(.NOT.SWITCH(I)) GO TO 6660
72  IF(IWRITE.EQ.1) CLOSE(20)
C W1.1
C INITIALIZE SUBROUTINE VARIABLES
C
6660 CALL RESET
IF(NFLAG(6).NE.1) GO TO 4
CALL RESET69
BLKBX=.TRUE.
GO TO 77
IF(NFLAG(4).NE.1) GO TO 5
SUB=.TRUE.
BLKBX=.TRUE.
IFLP=1
GO TO 78
C W1.2
C GEOMETRY INPUTS SETUP
C
5 BLKBX=.FALSE.
DISPLAY GEO
CALL RESET69
6 CALL RESET
LABEL(GIN)
ZSET(0.0F)
WRITE(16,108) NMB(1),NMB(2),IDPY,IFILE(1),IFILE(2),RHO
108 FORMAT(6B8X60S0Y030",2A5,
*"8X050GY004",14,
*"8X050GY040",2A5,
*"8X050GY004",F10.3)
WRITE(16,109) H,FACL,FACI,FACW
109 FORMAT(6B8X60S0Y070",F10.3,
*"8X050GY054",F10.3,
*"8X050GY060",F10.3,
*"8X050GY064",F10.3)
WRITE(16,110) FACG,WT,F,IREFL,WRITE,SUB
110 FORMAT(6B8X60S0Y070",F10.3,
*"8X050GY074",F10.3,
*"8X140GY030 ",I1,
*"8X140GY034 ",I1,
*"8X140GY040",L9)
WRITE(16,111) DELX(1),DELY,DELZ,PSI
111 FORMAT(6B8X60S0Y044",F10.3,
*"8X140GY050",F10.3,
*"8X140GY054",F10.3,
*"8X140GY060",F10.3)
WRITE(16,112) THETA,PHI,IBASE
112 FORMAT(6B8X60S0Y064",F10.3,
*"8X140GY070",F10.3,
*"8X140GY074 ",I1)
ENDLIST
15 LABEL(TEST3)
ZSET(0.0F)
IF(.NOT.IFLAG(30)) GO TO 99
I=1
IF(1FLAG(31)) I=4
WRITE(16,92) NMDS(I),NMDS(I+1),NMDS(I+2)
67  92 FORMAT("0B&X060&Y056",3A5)
70   89 IF(IFLAG(31)) GO TO 96
71       IF(LFLG(1).EQ.1) GO TO 88
72          WRITE(16,90) INUM
73     90 FORMAT("0B&X070&Y064",I10)
74          GO TO 93
75  88 WRITE(16,91) RRNUM
76  91 FORMAT("0B&X070&Y064",F10.5)
77          GO TO 93
100  96 WRITE(16,97) NME(1),NME(2)
101  97 FORMAT("0B&X070&Y064",2A5)
102  93 ENDLIST
103     IF(.NOT.IFLAG(30)) GO TO 15
104    CALL AN7
105     IF(.NOT.IFLAG(3)) GO TO 320
106        IF(LFLG(3).NE.1) GO TO 15
107         ENCODE(5,1700,NM6(1)) NME(1)
108         ENCODE(5,1700,NM8(2)) NME(2)
110        GO TO 6
112  320 IF(.NOT.IFLAG(4)) GO TO 321
113       IF(LFLG(3).NE.1) GO TO 15
1 IDPV=IINUM
2 IIPV=IOCT(IINUM)
3 GO TO 6
4 321 IF(.NOT.IFLAG(5)) GO TO 322
5 IF(LFLG(3).NE.1) GO TO 15
6 ENCODE(5,1700,IFILE(I)) NME(1)
7 ENCODE(5,1700,IFILE(2)) NME(2)
8 IGNME(1,NTOT+1)=IFILE(I)
9 IGNME(2,NTOT+1)=IFILE(2)
10 DO 3200 I=1,3
11 3200 IFILEN(I)=IFILE(I)
12 CALL LOCFILE(IIPV,IFILEN,IFILNO)
13 GO TO 6
14 322 IF(.NOT.IFLAG(6)) GO TO 325
15 IF(LFLG(3).NE.1) GO TO 15
16 RHO=RRNUM
17 GO TO 6
18 325 IF(.NOT.IFLAG(7)) GO TO 326
19 IF(LFLG(3).NE.1) GO TO 15
20 H=RRNUM
21 GO TO 6
22 326 DO 327 I=1,4
23 J=I+7
24 IF(.NOT.IFLAG(J)) GO TO 327
25 IF(LFLG(3).NE.1) GO TO 15
26 FACT(I)=RRNUM
27 GO TO 6
28 327 CONTINUE
29 IF(.NOT.IFLAG(12)) GO TO 328
30 IF(LFLG(3).NE.1) GO TO 15
31 IREFL=IINUM
32 GO TO 6
33 328 IF(.NOT.IFLAG(13)) GO TO 329
34 IF(LFLG(3).NE.1) GO TO 15
35 IWRITE=IINUM
36 GO TO 6
37 329 IF(.NOT.IFLAG(14)) GO TO 330
38 IF(LFLG(3).NE.1) GO TO 15
39 SUB=.FALSE.
40 IF(NME(I).EQ.1HT) SUB=.TRUE.
41 GO TO 6
42 330 DO 331 I=1,3
43 J=I+14
44 IF(.NOT.IFLAG(J)) GO TO 331
45 IF(LFLG(3).NE.1) GO TO 15
46 DELX(I)=RRNUM
47 GO TO 6
48 331 CONTINUE
49 DO 332 I=8,10
50 J=I+10
51 IF(.NOT.IFLAG(J)) GO TO 332
52 IF(LFLG(3).NE.1) GO TO 15
53 FACT(I)=RRNUM
54 GO TO 6
332 CONTINUE
70 IF(.NOT.IFLAG(22)) GO TO 359
71 IF(LFLG(3).NE.1) GO TO 15
72 WTF=RNUM
73 GO TO 6
74 IF(.NOT.IFLAG(23)) GO TO 358
75 IF(LFLG(3).NE.1) GO TO 15
76 IBASE=IINUM
77 GO TO 6
100 IF(IFLAG(21)) GO TO 77
101 GO TO 15
102 C W1.3
103 C BLACK BOX INPUTS SETUP
104 C
105 360 IFLP=0
106 CALL RESET69
107 IFILE(1)=IGNME(1,NO)
108 IFILE(2)=IGNME(2,NO)
110 DISPLAY BKBX
112 CALL RESET
113 LABEL(BIN)
114 ZSET(0.0F)
115 WRITE(16,208) NMB(1),NMB(2),(WTCI),I=1,7)
116 208 FORMAT("\$(5X00300Y0030",2A5,
117 * ",F10.0,
120 * ",F10.0,
* "@X@1210Y9040",F10.0,
2 * "@X@12170Y9040",F10.0,
3 * "@X@02030Y9044",F10.0,
4 * "@X@121100Y9044",F10.0,
5 * "@X@121700Y9044",F10.0
6 WRITE(16,212) (WT(I),I=8,13)
7 212 FORMAT("@B@X@00308Y9050",F10.0,
9 * "@X@121100Y9050",F10.0,
11 * "@X@121700Y9050",F10.0,
12 * "@X@020300Y9054",F10.0,
13 * "@X@121100Y9054",F10.0,
14 * "@X@121700Y9054",F10.0)
15 WRITE(16,209) FACL,DELX(I),FACI,DELY
16 209 FORMAT("@B@X@05008Y9064",F10.0,
17 * "@X@021400Y9064",F10.0,
20 * "@X@020500Y9070",F10.0,
21 * "@X@021400Y9070",F10.0)
22 WRITE(16,210) FACW,DELZ,FACG,PSI
23 210 FORMAT("@B@X@05008Y9074",F10.0,
24 * "@X@021400Y9074",F10.0,
25 * "@X@020500Y9070",F10.0,
26 * "@X@021400Y9070",F10.0)
27 WRITE(16,211) OVL,THETA,IWRITE,PHI,IPRINT,IBASE,IFILE(I),IFILE(2)
28 211 FORMAT("@B@X@05008Y9104 ",,03,
32 * "@X@021400Y9104",F10.0,
32 * "@X@020500Y9110",F10.0,
34 * "@X@021400Y9110",F10.0,
35 * "@X@020500Y9114",F10.0,
36 * "@X@021400Y9114",F10.0)
37 ENDLIST
39 798 LABEL(TEST3)
41 ZSET(0.0F)
42 IF(.NOT.IFLAG(30)) GO TO 213
43 I=1
44 IF(31) I=4
45 WRITE(16,214) NMDS(I),NMDS(I+1),NMDS(I+2)
46 214 FORMAT("@B@X@05008Y9056",2A5)
47 213 IF(IFLAG(31)) GO TO 220
48 IF(LFLG(1).EQ.1) GO TO 215
49 WRITE(16,216) IINUM
50 215 WRITE(16,217) RRNUM
51 216 FORMAT("@B@X@07008Y9064",I10)
52 GO TO 218
54 218 WRITE(16,219) RRNUM
55 219 IF(.NOT.IFLAG(30)) GO TO 219
56 GO TO 218
58 220 WRITE(16,221) NME(I),NME(2)
60 221 FORMAT("@B@X@07008Y9064",2A5)
61 222 ENDLIST
62 CALL AN7
64 IF(.NOT.IFLAG(29)) GO TO 799
65 IF(LFLG(3).NE.1) GO TO 798
66 ENCODE(5,1700,NME(1)) NME(1)
ENCLOSED: NMB(2) NME(2)

70       GO TO 797
71       DO 800 I=1,13
72       J=I+2
73       IF(.NOT.IFLAG(J)) GO TO 800
74       IF(LFLG(3).NE.1) GO TO 798
75       WT(I)=RRNUM
76       GO TO 797
77       800 CONTINUE
100      DO 805 I=1,4
101      J=I+15
1 IF(.NOT.IFLAG(J)) GO TO 805
2 IF(LFLG(3).NE.1) GO TO 798
3 FACT(I)=RRNUM
4 GO TO 797
5 805 CONTINUE
6 IF(.NOT.IFLAG(20)) GO TO 810
7 IF(LFLG(3).NE.1) GO TO 798
8 SUB=.FALSE.
9 IF(NME(1).EQ.1HT) SUB=.TRUE.
10 GO TO 797
11 810 IF(.NOT.IFLAG(21)) GO TO 815
12 IF(LFLG(3).NE.1) GO TO 798
13 IWRITE=IINUM
14 IF(.NOT.IFLAG(29)) GO TO 315
15 J=I+21
16 IF(.NOT.IFLAG(J)) GO TO 920
17 FACT(I)=RRNUM
18 GO TO 797
19 820 CONTINUE
20 DO 820 I=1,3
21 J=I+21
22 IF(.NOT.IFLAG(J)) GO TO 820
23 IF(LFLG(3).NE.1) GO TO 798
24 DELX(I)=RRNUM
25 GO TO 797
26 825 CONTINUE
27 DO 825 I=8,10
28 J=I+17
29 IF(.NOT.IFLAG(J)) GO TO 825
30 IF(LFLG(3).NE.1) GO TO 798
31 FACT(I)=RRNUM
32 GO TO 797
33 826 CONTINUE
34 IF(.NOT.IFLAG(32)) GO TO 826
35 IF(LFLG(3).NE.1) GO TO 798
36 IPRINT=IINUM
37 NUMO=9
38 IF(IPRINT.EQ.1) NUMO=13
39 GO TO 797
40 828 CONTINUE
41 IF(.NOT.IFLAG(34)) GO TO 827
42 IF(LFLG(3).NE.1) GO TO 798
43 ENCODE(S,1700,IFILE(1)) NME(1)
44 ENCODE(S,1700,IFILE(2)) NME(2)
45 ENCODE(S,1700,IGNME(1,NO)) NME(1)
46 ENCODE(S,1700,IGNME(2,NO)) NME(2)
47 GO TO 797
48 827 IF(.NOT.IFLAG(28)) GO TO 77
49 GO TO 798
50 C W1.4
51 C COMPONENT COUNTER
52 C
53 77 NTIME=NTIME+1
54 78 IF(NUMO.EQ.9) CALL $PTCH9
67  C ZERO ARRAYS
70  C
71     DO 6001 I=1,15
72     TEMP(I)=0.
73     SIXX(I)=0.
74  6001 CONTINUE
75  C W1.6
76  C SET UP DC MATRIX FOR DATA TO REFERENCE COORD SYSTEM
77  C
100    CALL XFORM(PSI, THETA, PHI, D2R)
101  C
1  IF(BLKBX) GO TO 7000
2  C
3  C W1.7
4  C READ IN ALL SURFACE DATA
5     CALL $NHALT
6  NUMI=20
7     OPEN(20, IIPY, IFILNO)
8  C READ BLANK CARD IN GEOMETRY FILE
9     READ(NUMI, 1550) IDUMMY
10    1550 FORMAT(I1)
11  C
12    READ(NUMI,1500,END=400) X,Y,Z,STAT, XX,YY,ZZ,STATT
13     RFLAG = .FALSE.
14    GO TO 80
15    30 IF (RFLAG) GO TO 50
RFLAG = .TRUE.
X = XX
Y = YY
Z = ZZ
STAT = STAT
GO TO 60
5 RFLAG = .FALSE.
READ (NUM+,1500,END=400) X,Y,Z,STAT, XX,YY,ZZ,STAT
60 IF (STAT .EQ. 0 .OR. STAT .EQ. 3) GO TO 180
12 IF (STAT .EQ. 2) GO TO 200
13 IF (.NOT. AFLAG) GO TO 200
MC = M
80 M = 1
IF (STAT .EQ. 2) GO TO 150
IF (.NOT. BFLAG) GO TO 84
75 DO 61 J = 1,MC
  61 XA(J) = XB(J)
  62 YA(J) = YB(J)
  63 ZA(J) = ZB(J)
83 XB(1) = X
  84 YB(1) = Y
  85 ZB(1) = Z
  86 GO TO 30
84 IF (AFLAG) GO TO 85
  85 BFLAG = .TRUE.
  86 GO TO 75
150 AFLAG = .TRUE.
  160 M = M + 1
160 IF (AFLAG) GO TO 160
  163 YA(M) = Y
  164 ZA(M) = Z
  165 GO TO 30
180 XA(M) = X
  185 YB(M) = Y
  186 ZB(M) = Z
  188 IF (STAT .NE. 3) GO TO 30
200 MMIN = MINO (MC,M) - 1
  205 MC = M
C W1 8
C BEGIN COMPUTATION OF SURFACE ELEMENT CHARACTERISTICS
C
DO 2000 J = 1,MMIN
  200 IPASS=0
  201 XIN(1) = +DELC(1)+XA(J)
  202 XIN(2) = +DELC(1)+YA(J+1)
  203 XIN(3) = +DELC(1)+XB(J+1)
  204 XIN(4) = +DELC(1)+X(J+1)
  205 YIN(1) = +DELY+YA(J)
  206 YIN(2) = +DELY+ZB(J+1)
1 \quad \text{YIN}(3) = \text{DELY} + \text{YB}(J + 1)
2 \quad \text{YIN}(4) = \text{DELY} + \text{YB}(J)
3 \quad \text{ZIN}(1) = \text{DELZ} + \text{ZA}(J)
4 \quad \text{ZIN}(2) = \text{DELZ} + \text{ZA}(J + 1)
5 \quad \text{ZIN}(3) = \text{DELZ} + \text{ZB}(J + 1)
6 \quad \text{ZIN}(4) = \text{DELZ} + \text{ZB}(J)

7 \quad \text{C TRANSFORM DATA TO REFERENCE COORD SYSTEM}

8 \quad \text{L1} (1) = \text{XIN}(4) - \text{XIN}(1)
9 \quad \text{L1} (2) = \text{YIN}(4) - \text{YIN}(1)
10 \quad \text{L1} (3) = \text{ZIN}(4) - \text{ZIN}(1)

11 \quad \text{C}
12 \quad \text{L2} (1) = \text{XIN}(2) - \text{XIN}(1)
13 \quad \text{L2} (2) = \text{YIN}(2) - \text{YIN}(1)
14 \quad \text{L2} (3) = \text{ZIN}(2) - \text{ZIN}(1)

15 \quad \text{C}
16 \quad \text{D1} (1) = \text{XIN}(4) - \text{XIN}(2)
17 \quad \text{D1} (2) = \text{YIN}(4) - \text{YIN}(2)
18 \quad \text{D1} (3) = \text{ZIN}(4) - \text{ZIN}(2)

19 \quad \text{C}
20 \quad \text{X0} (1) = \text{XIN}(1)
21 \quad \text{X0} (2) = \text{YIN}(1)
22 \quad \text{X0} (3) = \text{ZIN}(1)
23 \quad \text{GO TO 5010}

24 \quad \text{C}
25 \quad 5020 \quad \text{CONTINUE}
26 \quad \text{L1} (1) = \text{XIN}(2) - \text{XIN}(3)
27 \quad \text{L1} (2) = \text{YIN}(2) - \text{YIN}(3)
28 \quad \text{L1} (3) = \text{ZIN}(2) - \text{ZIN}(3)

29 \quad \text{C}
30 \quad \text{L2} (1) = \text{XIN}(4) - \text{XIN}(3)
31 \quad \text{L2} (2) = \text{YIN}(4) - \text{YIN}(3)
32 \quad \text{L2} (3) = \text{ZIN}(4) - \text{ZIN}(3)

33 \quad \text{C}
34 \quad \text{D1} (1) = \text{XIN}(2) - \text{XIN}(4)
35 \quad \text{D1} (2) = \text{YIN}(2) - \text{YIN}(4)
36 \quad \text{D1} (3) = \text{ZIN}(2) - \text{ZIN}(4)

37 \quad \text{C}
38 \quad \text{X0} (1) = \text{XIN}(3)
39 \quad \text{X0} (2) = \text{YIN}(3)
40 \quad \text{X0} (3) = \text{ZIN}(3)
41 \quad \text{GO TO 5010}

42 \quad \text{C}
43 \quad 5010 \quad \text{CONTINUE}
44 \quad \text{IPASS} = \text{IPASS} + 1

45 \quad \text{C}
46 \quad \text{CALL MXV(D2R,L1,L1)}
47 \quad \text{CALL MXV(D2R,L2,L2)}
CALL MXV(D2R,D1,D1)
CALL MXV(D2R,X0,X0)

CALL CROSS(D1,L1,ZU,ZM)
CALL CROSS(ZU,L1,YU,YM)

XM=SQRT(L1(1)**2+L1(2)**2+L1(3)**2)

IF(XM.LE.0.) GO TO 2000

DO 5000 I=1,3

XU(I)=L1(I)/XM
YU(I)=YU(I)/YM
ZU(I)=ZU(I)/ZM

DC(3*I-2)=XU(I)
DC(3*I-1)=YU(I)
DC(3*I)=ZU(I)

5000 CONTINUE

CALL MXV(DC,L2,TEMP)
A=TEMP(2)
B=XM
C=TEMP(1)

S=A*B/2.

IF(S.LE.0.) GO TO 2000

WT(I)=S*RHO*H

C W1.10

C CENTROID OF TRIANGLE

XBAR(1)=(B+C)/3.
XBAR(2)=A/3.
XBAR(3)=0.

C W1.11

C INERTIA MATRIX IN PANEL COORDINATES

TMP=WT(1)/18.

PMIP(1)=TMP*A**2
PMIP(5)=TMP*(B**2-B*C+C**2)
PMIP(9)=PMIP(1)+PMIP(5)
PMIP(2)=-TMP*A*(B/2.-C)
PMIP(4)=PMIP(2)
PMIP(3)=0.
PMIP(6)=0.
PMIP(7)=0.
PMIP(8)=0.

C W1.12

C ROTATE INERTIA MATRIX TO PARALLEL TO REFERENCE FRAME

CALL MXM3X3(PMIP,DC,TEMP)
CALL XPOSE(DC,TEMP2)
CALL MXM3X3(TEMP2,TEMP,PMI)

C W1.13

C COMPUTE CENTROID POSITION IN REFERENCE FRAME

C

CALL MXV(TEMP2,XBAR,TEMP)

XBRF(1)=TEMP(1)+X0(1)
XBRF(2)=TEMP(2)+X0(2)
XBRF(3)=TEMP(3)+X0(3)
C W1.14
VOLUME OF PANEL WRT X-Y PLANE AND AREA ON Y-Z PLANE

VOL=DC(9)*S*XBFRF(3)
AREA=S*DC(3)
IF(AREA.LT.0.) AREA=0.

C W1.15
MOMENT OF PANEL IN REF COORD

DO 5100 N=1,3
WT(N+1)=XBFRF(N)*WT(1)
5100 CONTINUE

C W1.16
COMPUTE MOMENTS AND PRODUCTS OF INERTIA IN REFERENCE FRAME

IXX=PMI(1)
IYY=PMI(5)
IZZ=PMI(9)
IXY=-PMI(2)
IXZ=-PMI(3)
IYZ=-PMI(6)
CALL PAT(IXX,WT(1),XBFRF,WT(5))

C W1.17
SUBTOTAL OF MASS PROPERTIES

CALL TOT(SIXX,WT,15)

C W1.18
IS QUADRALATERAL COMPLETED?

IF(IPASS.EQ.1) GO TO 5020

GO TO 60

GO TO 400 CONTINUE

WRITE(10,6941)
6941 FORMAT( 'MADE IT TO 400')

C W1.19
SET DELTA CG'S FOR NEW BASE LINE

IF(IBASE.EQ.0) GO TO 4041

DO 4040 I=1,3
DCG(NO,I)=0.
4040 CONTINUE

IBASE=0

C W1.20
START SUMMATION OF BLACK BOXES

C

IRF=1
4041 CALL LOAD(SIXX,TEMP,15)

IF(TEMP(1).LT.1.E-4) GO TO 402

DO 405 I=2,4
J=I+1
IF(ABS(TEMP(I)).LT.1.) TEMP(I)=0.
TEMP(J)=TEMP(I)
405 TEMP(I)=TEMP(I)*TEMP(1)

CALL PAT(TEMP(5),TEMP(1),TEMP(2),TEMP(5))
67 TEMP(5) = TEMP(5) - 2 * (TEMP(17) * TEMP(3) + TEMP(18) * TEMP(4))
70 TEMP(6) = TEMP(6) - 2 * (TEMP(16) * TEMP(2) + TEMP(18) * TEMP(4))
71 TEMP(7) = TEMP(7) - 2 * (TEMP(16) * TEMP(2) + TEMP(17) * TEMP(3))
72 TEMP(8) = TEMP(8) - TEMP(16) * TEMP(3) - TEMP(17) * TEMP(2)
73 TEMP(9) = TEMP(9) - TEMP(16) * TEMP(4) - TEMP(18) * TEMP(2)
74 TEMP(10) = TEMP(10) - TEMP(17) * TEMP(4) - TEMP(18) * TEMP(3)
75 402 IF (IREFL.EQ.0 .OR. IRS.GT.1) GO TO 401
76 TEMP(3) = -TEMP(3)
77 TEMP(8) = -TEMP(8)
100 TEMP(10) = -TEMP(10)
101 CALL PAT(TEMP(5), TEMP(1), TEMP(2), TEMP(5))
1 TEMP(2)=TEMP(2)*TEMP(1)
2 TEMP(3)=TEMP(3)*TEMP(1)
3 TEMP(4)=TEMP(4)*TEMP(1)
4 CALL TOT(SIXX,TEMP,15)
5 IRF=2
6 GO TO 403
7 401 CONTINUE
8 IF<(BLKBY).OR.(WT,EQ.0.).OR.(IRF,EQ.3)) GO TO 469
9 C W1.21
10 DENSITY CALCULATION FOR GEOMETRY OPTION
11 C
12 RHO=WTF/(TEMP(11)*H)
13 SIXX(I)=TEMP(I)=WT
14 DO 470 I=5,10
15 SIXX(I)=TEMP(I)=TEMP(I)*RHO
16 470 CONTINUE
17 469 CONTINUE
18 DO 404 I=5,10
19 404 TEMP(I)=TEMP(I)/4633.056
20 FAC2=FAACL*FAACL
21 TEMP(11)=TEMP(11)/FAC2
22 TEMP(13)=TEMP(13)/FAC2
23 TEMP(12)=TEMP(I2)/(FAC2*FAACL)
24 IF(IWRITE.EQ.0) GO TO 4999
25 C W1.22
26 WRITE RESULTS TO DISK
27 C
28 WRITE(20,1000) NMB(1),NMB(2),(TEMP(I),I=1,10)
29 WRITE(20,1011) (TEMP(I),I=11,13)
30 4999 IF<(NFLAG(6).EQ.1).OR.(IFLP.EQ.1)) GO TO 410
31 C W1.23
32 WRITE RESULTS TO ANK OR PPL
33 C
34 WRITE(NUM0,1000) NMB(1),NMB(2),(TEMP(I),I=1,10)
35 WRITE(NUM0,1011) (TEMP(I),I=11,13)
36 410 IF<(NFLAG(6).EQ.1).OR.(IFLP.EQ.1)) GO TO 410
37 IF<(NFLAG(6).EQ.1).AND.(IRF.EQ.3)) GO TO 410
38 C W1.24
39 SUBTRACT COMPONENT'S MASS PROP, FROM TOTALS
40 C
41 WRITE(10,6991) NFLAG(6)
42 6991 FORMAT(' NFLAG(6)=',I3)
43 IF(NFLAG(6).EQ 1) RETURN
44 IF(IFLP.EQ.1.) GO TO 360
45 WRITE(10,6991) NFLAG(6)
46 6991 FORMAT(' NFLAG(6)=',I3)
47 IF(NFLAG(6).EQ 1) RETURN
48 RETURN
49 DO JJ=1,13
50 L 1.24
51 SUBTRACT COMPONENT'S MASS PROP, FROM TOTALS

334  $\text{SIXX(JJ)} = \text{SIXX(JJ)}$

333  CONTINUE

CALL TOT(TIXX,SIXX,15)

IF(NFLAG(4).NE.1) GO TO 335

NAME(1,NO)=NMB(1)

NAME(2,NO)=NMB(2)

335  IF(IFLP.EQ.1) GO TO 336

ROOT=STNAME

ENCODE(5,1700,NMB(1)) ROOT

ENCODE(5,1710,NMB(2)) NTIME

336  IF(I=3)

CALL LOAD(TIXX,TEMP,15)

GO TO 409

C  W1.25

C  BLACKBOX CONTRIBUTIONS

C

?000 CONTINUE

CALL LOAD(WT,SIXX,15)

DO 7010 I=1,10

FAC=FACI*4633.056

IF(I.LT.5) FAC=FACG

IF(I.EQ.1) FAC=FACW

SIXX(I)=SIXX(I)*FAC

7010 CONTINUE

SIXX(2)=SIXX(2)+DELX(1)

SIXX(3)=SIXX(3)+DELY

SIXX(4)=SIXX(4)+DELZ

CALL MXV(D2R,SIXX(2),SIXX(2))

PMIP(1)=SIXX(5)

PMIP(5)=SIXX(6)

PMIP(9)=SIXX(7)

PMIP(2)=-SIXX(8)

PMIP(4)=-SIXX(8)

PMIP(3)=-SIXX(9)

PMIP(7)=-SIXX(9)

PMIP(6)=-SIXX(10)

PMIP(8)=-SIXX(10)

CALL XPOSE(D2R,TEMP)

CALL MXM3X3(PMIP,TEMP,TEMP2)

CALL MXM3X3(D2R,TEMP2,PMI)
1  SIXX(5)=PMI(1)
2  SIXX(6)=PMI(5)
3  SIXX(7)=PMI(9)
4  SIXX(8)=PMI(2)
5  SIXX(9)=PMI(3)
6  SIXX(10)=PMI(6)
7  CALL PAT(SIXX(5),SIXX(1),SIXX(2),SIXX(5))
10 SIXX(2)=SIXX(2)*SIXX(1)
11 SIXX(3)=SIXX(3)*SIXX(1)
12 SIXX(4)=SIXX(4)*SIXX(1)
13 WRITE(10,6900)
14 6900 FORMAT(' GO TO 400')
15 GO TO 400
16 IMAGE GEO
17 C W2.0
20 C DISPLAY GEOMETRY INPUTS
21 C
22 CLEARPEN(1)
23 CHAR("@B6X055@Y020GEOMETRY INPUTS")
24 CHAR(GIN)
25 PENON(1)
26 MENU("@B6X020@Y030NAME =9.9A8")
* 8X90206Y9034INPUT PVV = .8A9
2 * 8X9020Y9040GEOM FILE = .8A9
3 * 8X90206Y9044RHO = .8A9
4 * 8X90206Y9050H = .8A9
5 * 8X90206Y9054FACL = .8A9
6 * 8X90206Y9060FACI = .8A9
7 * 8X90206Y9064FACW = .8A9
10 * 8X90206Y9070FACG = .8A9
11 * 8X90206Y9074FIX WT = .8A9
12 * 8X91106Y9030IREFL = .8A9
13 * 8X91106Y9034IWRITE = .8A9
14 * 8X91106Y9040SUB = .8A9
15 * 8X91106Y9044DELX = .8A9
16 * 8X91106Y9050DELY = .8A9
17 * 8X91106Y9054DELZ = .8A9
20 * 8X91106Y9060PSI = .8A9
21 * 8X91106Y9064THETA = .8A9
22 * 8X91106Y9070PHI = .8A9
23 * 8X91106Y9074BASE CHNGE = .8A9
24 * 8X90706Y8104PROCEED,$2"
25 PENOFF(1)
26 GO TO (300, 301, 302, 303, 304, 305, 306, 307, 308, 298, 309),
27 * 310, 311, 312, 313, 314, 315, 316, 317, 297, 299)
30 GO TO 500
31 300 IFLAG(3) = .TRUE.
32 IFLAG(31) = .TRUE.
33 IFLAG(30) = .TRUE.
34 GO TO 500
35 301 IFLAG(4) = .TRUE.
36 IFLAG(30) = .TRUE.
37 GO TO 500
40 302 IFLAG(5) = .TRUE.
41 IFLAG(31) = .TRUE.
42 IFLAG(30) = .TRUE.
43 GO TO 500
44 303 IFLAG(6) = .TRUE.
45 IFLAG(30) = .TRUE.
46 GO TO 500
47 304 IFLAG(7) = .TRUE.
50 IFLAG(30) = .TRUE.
51 GO TO 500
52 305 IFLAG(8) = .TRUE.
53 IFLAG(30) = .TRUE.
54 GO TO 500
55 306 IFLAG(9) = .TRUE.
56 IFLAG(30) = .TRUE.
57 GO TO 500
60 307 IFLAG(10) = .TRUE.
61 IFLAG(30) = .TRUE.
62 GO TO 500
63 308 IFLAG(11) = .TRUE.
64 IFLAG(30) = .TRUE.
65 GO TO 500
66 309 IFLAG(12) = .TRUE.
67       IFLAG(30) = .TRUE.
70       GO TO 500
71  310   IFLAG(13) = .TRUE.
72  311   IFLAG(14) = .TRUE.
73       GO TO 500
74  312   IFLAG(15) = .TRUE.
75  313   IFLAG(16) = .TRUE.
76       GO TO 500
77  314   IFLAG(17) = .TRUE.
78       GO TO 500
79  315   IFLAG(18) = .TRUE.
80       GO TO 500
81  316   IFLAG(19) = .TRUE.
82       GO TO 500
83  317   IFLAG(20) = .TRUE.
84       GO TO 500
85  318   IFLAG(21) = .TRUE.
86       GO TO 500
87  319   IFLAG(22) = .TRUE.
88       GO TO 500
89  320   IFLAG(23) = .TRUE.
90       GO TO 500
91  321   IFLAG(24) = .TRUE.
92       GO TO 500
93  322   IFLAG(25) = .TRUE.
94       GO TO 500
95  323   IFLAG(26) = .TRUE.
96       GO TO 500
97  324   IFLAG(27) = .TRUE.
98       GO TO 500
99  325   IFLAG(28) = .TRUE.
100  326   IFLAG(29) = .TRUE.
101  327   CALLSIM GRIDW
102       RETURN
103  328   IMAGE BKBOX
104       CLEARPEN(1)
105       CHAR("0B9X0606Y02OBLACK BOX INPUTS")
1       CHAR(BIN)
2       PENOFF(1)
3       MENU("@x010gY0034lNAME= 9.8A0
4 * @x010gY0040XCG = 9.8A0
5 * @x040gY0040YCG = 9.8A0
6 * @x0010gY0040ZCG = 9.8A0
7 * @x010gY0044lXX = 9.8A0
8 * @x0070gY0044lYY = 9.8A0
9 * @x150gY0044lZZ = 9.8A0
10 * @x010gY0050lXY = 9.8A0
11 * @x0070gY0050lYZ = 9.8A0
12 * @x150gY0050lXZ = 9.8A0
13 * @x010gY0054lASF = 9.8A0
14 * @x0070gY0054lVOL = 9.8A0
15 * @x150gY0054lAFR = 9.8A0
16 * @x020gY0064lFACL = 9.8A0
17 * @x020gY0070lFACI = 9.8A0
18 * @x020gY0074lFACW = 9.8A0
19 * @x020gY0100lFACG = 9.8A0
20 * @x020gY0104lVOL = 9.8A0
21 * @x020gY1100lWRITE = 9.8A0
22 * @x020gY1114lPRINT = 9.8A0
23 * @x020gY1200lGEOM FILE = 9.8A0
24 * @x110gY0064lDELX = 9.8A0
25 * @x110gY0070lDELY = 9.8A0
26 * @x110gY0074lDELZ = 9.8A0
27 * @x110gY1000lPSI = 9.8A0
28 * @x110gY1004lTHETA = 9.8A0
29 * @x110gY1100lPHI = 9.8A0
30 * @x110gY1114lBASE CHNGE= 9.8A0
31 * @x070gY130PROCEED 0.92")
32 PENOFF(1)
33 GO TO (364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377,
34 378, 379, 380, 381, 382, 383, 391, 397, 384, 385, 396, 387, 388, 389, 392, 390
35 )
36 GO TO 393
37 364 IFLAG(29)=.TRUE.
38 IFLAG(30)=.TRUE.
39 IFLAG(31)=.TRUE.
40 GO TO 393
41 365 IFLAG(3)=.TRUE.
42 IFLAG(30)=.TRUE.
43 GO TO 393
44 366 IFLAG(4)=.TRUE.
45 IFLAG(30)=.TRUE.
46 GO TO 393
47 367 IFLAG(5)=.TRUE.
48 IFLAG(30)=.TRUE.
49 GO TO 393
50 368 IFLAG(6)=.TRUE.
51 IFLAG(30)=.TRUE.
52 GO TO 393
53 369 IFLAG(7)=.TRUE.
54 IFLAG(30)=.TRUE.
55 GO TO 393
56 370 IFLAG(8)=.TRUE.
57 IFLAG(30)=.TRUE.
58 GO TO 393
59 371 IFLAG(9)=.TRUE.
60 IFLAG(30)=.TRUE.
61 GO TO 393
62 372 IFLAG(10)=.TRUE.
63 IFLAG(30)=.TRUE.
64 GO TO 393
65 373 IFLAG(11)=.TRUE.
66 IFLAG(30)=.TRUE.

C-2
67  GO TO 393
70  370  IFLAG(8)=.TRUE.
71  IFLAG(30)=.TRUE.
72  GO TO 393
73  371  IFLAG(9)=.TRUE.
74  IFLAG(30)=.TRUE.
75  GO TO 393
76  372  IFLAG(10)=.TRUE.
77  IFLAG(30)=.TRUE.
100 GO TO 393
101  373  IFLAG(11)=.TRUE.
102  IFLAG(30)=.TRUE.
103  GO TO 393
104  374  IFLAG(12)=.TRUE.
105  IFLAG(30)=.TRUE.
106  GO TO 393
107  375  IFLAG(13)=.TRUE.
110  IFLAG(30)=.TRUE.
111  GO TO 393
IFLAG(14) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(15) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(16) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(17) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(18) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(19) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(20) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(21) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(22) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(23) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(24) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(25) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(26) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(27) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(28) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(29) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(32) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(33) = .TRUE.
IFLAG(30) = .TRUE.
GO TO 393

IFLAG(34) = .TRUE.
IFLAG(31) = .TRUE.
IFLAG(30)=.TRUE.

CALL SIMG GRIDW
RETURN

SUBROUTINE RESET
C W4.0
C INITIALIZE SUBROUTINE VARIABLES AFTER EACH INPUT
C
IINUM=0
RRNUM=0.
DO 395 I=1,40
IFLAG(I)=.FALSE.
RETURN
SUBROUTINE RESET69
C W5.0
C INITIALIZE SUBROUTINE VARIABLES AFTER EACH CALCULATION
C
IF(NFLAG(5),NE.1) GO TO 396
NMB(1)=5HDATA
NMB(2)=5HFILE
IDPV=110
IPV=110B
IFILE(1)=5H
IFILE(2)=5H
396 IREFL=IWRITE=IPRINT=0
WTF=DELX=DELY=DELZ PHI=THETA=PSI=0.
RH0=H=FACI=FACW=FACG=1.
FACL=12.
SUB=.FALSE.
RETURN
END
SUBROUTINE WABLST

C MENU COMPUTATION FOR COMPONENT DISPLAY

C COMMON/ PARA/ LFLG(10), MFLAG(4), SUM, LETTER, NO, NFLAG, IINO, NFLAG(12)
C * , JJ, KKK, LLL, MMM, NNN, NAME(2, 60), NME(2), INUM, RNUM
C COMMON/ CNT/ IPASS, NUP, NAMER(3), IPV, ISTRT, IEND, IT, LINES
C COMMON/ OPT/ NHIT, OPDPL
C COMMON/ STR/ NTOT, VALUE1(60), VALUE2(60), VALUE3(60), VALUE4(60),
C * , IPASS, NUP, NAMER(3), IPV, ISTRT, IEND, IT, LINES
C COMMON/ WB/ NMB(2), WTB(20), TOT1(20), TOT2(20), TOT3(20), TOT4(20)
C INTEGER MNUI(200), MNU2(200), MNU3(200), MNU4(200), MNU5(200)
C REAL VALUE1, VALUE2, VALUE3, VALUE4
C DATA NF/10/
C DATA MASK(8*)1 /
C 1234567890 1234567890 1234567890 1234567890 1234567890
C DATA FMT('/9H"&S8B9B0.03,3H@Y0.03,6H@8A")'/
C IF(NUP.EQ.1) GO TO 100
C IF(IPASS.NE.0) GO TO 900
C IPASS=IPASS+1
C L1.1

C COMPONENT'S NUMBER MENU

C
C L1.2

C COMPONENT'S NAME MENU

C

C 19ASL5T VER 1 REV A 16 JUN 76 PAGE 1
300 CONTINUE

C L1.3
C TOTALS MENU
C
7   DY=DY+4
10  LABEL(TOT1)
11  ZSET(0.0F)
12  ENCODE(40,FMT,MASK,H) DX,DY,ITNM(1),ITNM(2)
13  WRITE(16,MASK)
14  ENDLIST
15  C L1.4
16  C COMPONENT'S WEIGHT MENU
17  C
20  DY=24
21  DX=25
22  LABEL(MNU2)
23  ZSET(0.0F)
24  DO 400 J=ISTR1,IEND
25  FMT(8)=5H0A"
26  IF(J.EQ.IEND) FMT(8)=5H0Z"
27  FMT(6)=5H
28  CALL DECFMT(VALUE1(J),NF,FMT(6))
29  ENCODE(40,FMT,MASK,H) DX,DY,VALUE1(J)
30  DY=DY+4
31  WRITE(16,MASK)
32  CONTINUE
33  ENDLIST
34  C L1.5
35  C TOTAL WEIGHT MENU
36  C
39  DY=DY+4
40  LABEL(TOT2)
41  ZSET(0.0F)
42  ENCODE(40,FMT,MASK,H) DX,DY,TEMP(1)
43  WRITE(16,MASK)
44  ENDLIST
47  C L1.6
50  C COMPONENT'S X-CG MENU
51  C
52  DY=24
53  DX=40
54  LABEL(MNU3)
55  ZSET(0.0F)
56  DO 500 J=ISTR1,IEND
57  FMT(8)=5H0A"
58  IF(J.EQ.IEND) FMT(8)=5H0Z"
59  FMT(6)=5H
60  CALL DECFMT(VALUE2(J),NF,FMT(6))
63  ENCODE(40,FMT,MASK,H) DX,DY,VALUE2(J)
64  DY=DY+4
65  WRITE(16,MASK)
500 CONTINUE

98
67       ENDLIST
70      C   L1.7
71      C   TOTAL X-CG MENU
72      C
73       DY=DY+4
74       LABEL(TOT3)
75       ZSET(0.0F)
76       ENCODE(40,FMT,MASK,H) DX,DY,TEMP(2)
77       WRITE(16,MASK)
100     ENDLIST
101     C   L1.8
102     C   COMPONENT'S Y-CG MENU
103     C
104       DY=24
105       DX=55
106      LABEL(MNU4)
107      ZSET(0.0F)
110     DO 600 J=ISTRT,IEND
111    FMT(8)=SH02 ''
112    IF(J.EQ.IEND) FMT(8)=5H02''
113    FMT(6)=5H
114    CALL DECFMT(VALUE3(J),NF,FMT(6))
115    ENCODE(40,FMT,MASK,H) DX,DY,VALUE3(J)
116    DY=DY+4
117    WRITE(16,MASK)
600   CONTINUE
121     ENDLIST
122      C   L1.9
123      C   TOTAL Y-CG MENU
124      C
125       DY=DY+4
126      LABEL(TOT4)
127      ZSET(0.0F)
130    ENCODE(40,FMT,MASK,H) DX,DY,TEMP(3)
131    WRITE(16,MASK)
132     ENDLIST
133      C   L1.10
134      C   COMPONENT'S Z-CG MENU
135      C
136       DY=24
137       DX=70
140    LABEL(MNU5)
141    ZSET(0.0F)
DO 700 J=ISTRT,IEND
FMT(8)=5H6A"
IF(J.EQ.IEND) FMT(8)=5H82"
FMT(6)=5H
CALL DECFMT(VALUE4(J),NF,FMT(6))
ENCOD40,FMT,MASK,H) DX,DY,VALUE4(J)
DY=DY+4
WRITE(16,MASK)
700 CONTINUE
ENDLIST
C L1.11
C TOTAL Z-CG MENU
C
DY=DY+4
LABEL(TOT5)
ZSET(0.0F)
ENCODE40,FMT,MASK,H) DX,DY,TEMP(4)
WRITE16,MASK)
ENDLIST
NUP=0
800 DISPLAY WABMNU
900 IF(LFLG(3).NE.1) RETURN
IF(JJJ.EQ.0) GO TO 1015
NAME1,JJJ)=NME1)
NAME2,JJJ)=NME2)
NUP=1
RETURN
1015 IF(KKK.EQ.0) GO TO 1030
VALUE1(KKK)=RRNUM
NUP=1
RETURN
1030 IF(LLL.EQ.0) GO TO 1045
PWT1)=VALUE2(LLL)
VALUE2(LLL)=RRNUM
DCG(LLL,1)=DCG(LLL,1)+(VALUE2(LLL)-PWT1))
NUP=1
RETURN
1045 IF(MMM.EQ.0) GO TO 1060
PWT(2)=VALUE3(MMM)
VALUE3(MMM)=RRNUM
DCG(MMM,2)=DCG(MMM,2)+(VALUE3(MMM)-PWT(2))
NUP=1
RETURN
1060 IF(NNN.EQ.0) GO TO 800
PWT(3)=VALUE4(NNN)
1 VALUE4(NNN)=RRNUM
2 DCG(NNN,3)=DCG(NNN,3)+(VALUE4(NNN)-PWT(3))
3 NUP=1
4 RETURN
5 IMAGE WABMNNU
6 C L2.0
7 C DISPLAY COMPONENTS' MASS PROPERTIES
10 C
11 CHAR("S$080100Y5020 COMP WB X-CG")
12 * Y-CG Z-CG"
13 CHAR(MNU6)
14 CLEARPEN(1)
15 $PWD4=0
16 PENON(1)
17 MENU(MNU1)
20 PENOFF(1)
21 CHAR(TOT1)
22 IF(LFLG(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)
23 * .OR.(NHIT.EQ.1)) GO TO 1101
24 IF($PWD4.EQ.0) GO TO 1101
25 LFLG(6)=1
26 JJJ=NO=$PWD4*(IT*LINES)
27 MFLAG(1)=1
30 1101 CLEARPEN(1)
31 $PWD4=0
32 PENON(1)
33 MENU(MNU2)
34 PENOFF(1)
35 CHAR(TOT2)
36 IF(LFLG(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)
37 * .OR.(NHIT.EQ.1)) GO TO 1111
40 IF($PWD4.EQ.0) GO TO 1111
41 LFLG(6)=1
42 KKK=NO=$PWD4*(IT*LINES)
43 IF(KKK.NE.0) INO=2
44 MFLAG(2)=1
45 1111 CLEARPEN(1)
46 $PWD4=0
47 PENON(1)
50 MENU(MNU3)
51 PENOFF(1)
52 CHAR(TOT3)
53 IF(LFLG(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)
54 * .OR.(NHIT.EQ.1)) GO TO 1121
55 IF($PWD4.EQ.0) GO TO 1121
56 LFLG(6)=1
57 LLL=NO=$PWD4*(IT*LINES)
60 IF(LLL.NE.0) INO=3
61 MFLAG(2)=1
62 1121 CLEARPEN(1)
63 $PWD4=0
64 PENON(1)
65 MENU(MNU4)
66 PENOFF(1)
67            CHAR(TOT4)
70            IF(<LFLAG(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)
71                .OR.(NHIT.EQ.1)) GO TO 1131
72            IF(*PWD4.EQ.0) GO TO 1131
73            LFLG(6)=1
74            MMM=NO=$PWD4+(IT*LINES)
75            IF(MMM.NE.0) INO=4
76            MFLAG(2)=1
77            1131 CLEARPEN(1)
100            $PWD4=0
101            PENON(1)
102            MENU(MNU5)
103            PENOFF(1)
104            CHAR(TOT5)
105            IF(<LFLAG(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)
106                .OR.(NHIT.EQ.1)) GO TO 1200
107            IF($PWD4.EQ.0) GO TO 1200
110            LFLG(6)=1
111            NNN=NO=$PWD4+(IT*LINES)
112            IF(NNN.NE.0) INO=5
113            MFLAG(2)=1
114            1200 IF(<MFLAG(1).NE.0).OR.(MFLAG(2).NE.0)) NFLG=1
115            CALLSIMG OPTIONS
116            RETURN
117            END
SUBROUTINE WABSET
C
C ARRAY SETUP FOR WABCOM
C
COMMON/STR/ NTOT, VALUE(60), VALUE2(60), VALUE3(60), VALUE4(60),
* VALUE5(60), VALUE6(60), VALUE7(60), VALUE8(60), VALUE9(60),
* VALUE10(60), VALUE11(60), VALUE12(60), VALUE13(60)
COMMON/PARA/ LFLAG(10), MFLAG(4), SUM, LETTER, NO, NFLG, INO, NFLAG(12),
* JJJ, KKK, LLL, MMM, NNN, NAME(2, 60), NME(2), NINUM, RNNUM
COMMON/WB/ NMB(2), WT(15), TEMP(18), IWRITE, IJ, ITNM(2), TIXX(15), NUMO
* , IPRINT, PWT(3), DCG(60, 3), IGNME(3, 60)
DIMENSION V(60, 13)
EQUIVALENCE (V(1, 1), VALUE(1))
DO 50 I=1, 15
TIXX(I)=0.
50 CONTINUE
DO 100 IJ=1, NTOT
NMB(1)=NAME(1, IJ)
NMB(2)=NAME(2, IJ)
DO 200 IK=1, 13
WT(IK)=V(IJ, IK)
200 CONTINUE
CALL WABCOM
100 CONTINUE
IF(IWRITE.EQ.1) CLOSE(20)
ITNM(1)=SHTOTAL
ITNM(2)=SHS
RETURN
END
SUBROUTINE XFORM(A,B,C,X)

DESCRIPTION: Computes the transformation matrix for a yaw, pitch, roll rotation sequence.

INPUTS:
- A = Yaw angle in degrees
- B = Pitch angle in degrees
- C = Roll angle in degrees

OUTPUT:
- X = 3x3 matrix of transformation direction cosines

DIMENSION X(9)
SA = SIN(A/57.295779)
CA = COS(A/57.295779)
SB = SIN(B/57.295779)
CB = COS(B/57.295779)
SC = SIN(C/57.295779)
CC = COS(C/57.295779)

X(1) = CB*CA
X(2) = SC*SB+CA-CC*SA
X(3) = CC*SB*CA+SC*SA
X(4) = CB*SA
X(5) = SC*SB*SA+CC*CA
X(6) = CC*SB*SA-SC*CA
X(7) = -SB
X(8) = SC*CE
X(9) = CC*CD
RETURN
END
SUBROUTINE XPOSE(A,B)

B = TRANSPOSE OF A
A & B ARE 3X3 MATRICES

DIMENSION A(9), B(9)

B(1) = A(1)
B(2) = A(4)
B(3) = A(7)
B(4) = A(2)
B(5) = A(5)
B(6) = A(8)
B(7) = A(3)
B(8) = A(6)
B(9) = A(9)

RETURN

END