ALEX POPE

CR151896

SIGMA CORPORATION
TECHNICAL NOTE

JTN 76-109
VOLUME I

(NASA-CR-151896) IMP: INTERACTIVE MASS PROPERTIES PROGRAM. VOLUME 1: PROGRAM DESCRIPTION (Sigma Corp., Houston, Tex.) 111 p HC A06/MF A01 CSCL 09B

N79-17575

Unclas
G3/61 13898

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

LIST OF SYMBOLS

A	Area
A_p	Projected Area
B	Base of Triangular Element
BX	X-Projection of the Base of the Triangular Element
BY	Y-Projection of the Base of the Triangular Element
BZ	Z-Projection of the Base of the Triangular Element
C	Offset of the Vertex of a Triangular Surface Element
CV	Center of Volume
D	One side of a Triangular Surface Element
DC	Matrix of Direction Cosines of a Triangular Surface Element
DX	X-Projection of the D-Side of a Triangular Surface Element
DY	Y-Projection of the D-Side of a Triangular Surface Element
DZ	Z-Projection of the D-Side of a Triangular Surface Element
H	Thickness of a Triangular Surface Element
I	Element Index in a Section Plane
J	Element Index of Sections
I_{xx}	Moment of Inertia (X-Axis)
I_{yy}	Moment of Inertia (Y-Axis)
I_{zz}	Moment of Inertia (Z-Axis)
I_{xy}	Product of Inertia (X-Y Plane)
I_{yz}	Product of Inertia (Y-Z Plane)

I_{zx}	Product of Inertia (Z-X Plane)
PMI	Inertia Matrix Centered at a Triangular Element c.g.
PMIP	Inertia Matrix Centered at the Reference System Axes.
V	Enclosed Volume of Elements.
X	Measure of Distance in the X Direction
XO	X Coordinate of the Jth Section
XBR	Y Coordinate of the Base of the Triangular Element
XPX	Component of Unit Vector in the DC Matrix
XPY	Component of Unit Vector in the DC Matrix
XPZ	Component of Unit Vector in the DC Matrix
x	Local (Triangle) x-Axis
Y	Measure of Distance in the Y Direction
YO	Y Coordinate of (I, J) Element
YBR	Y Coordinate of the Base of the Triangular Element
YPX	Component of Unit Vector in the DC Matrix
YPY	Component of Unit Vector in the DC Matrix
YPZ	Component of Unit Vector in the DC Matrix
Y	Local (Triangle) y-Axis
Z	System Reference Axis
ZO	Z-Coordinate of (I, J) Element
ZBR	Z-Coordinate of the Base of the Triangular Element
zpx	Component of Unit Vector in the DC Matrix
ZPY	Component of Unit Vector in the DC Matrix
ZPZ	Component of Unit Vector in the DC Matrix
Z	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.

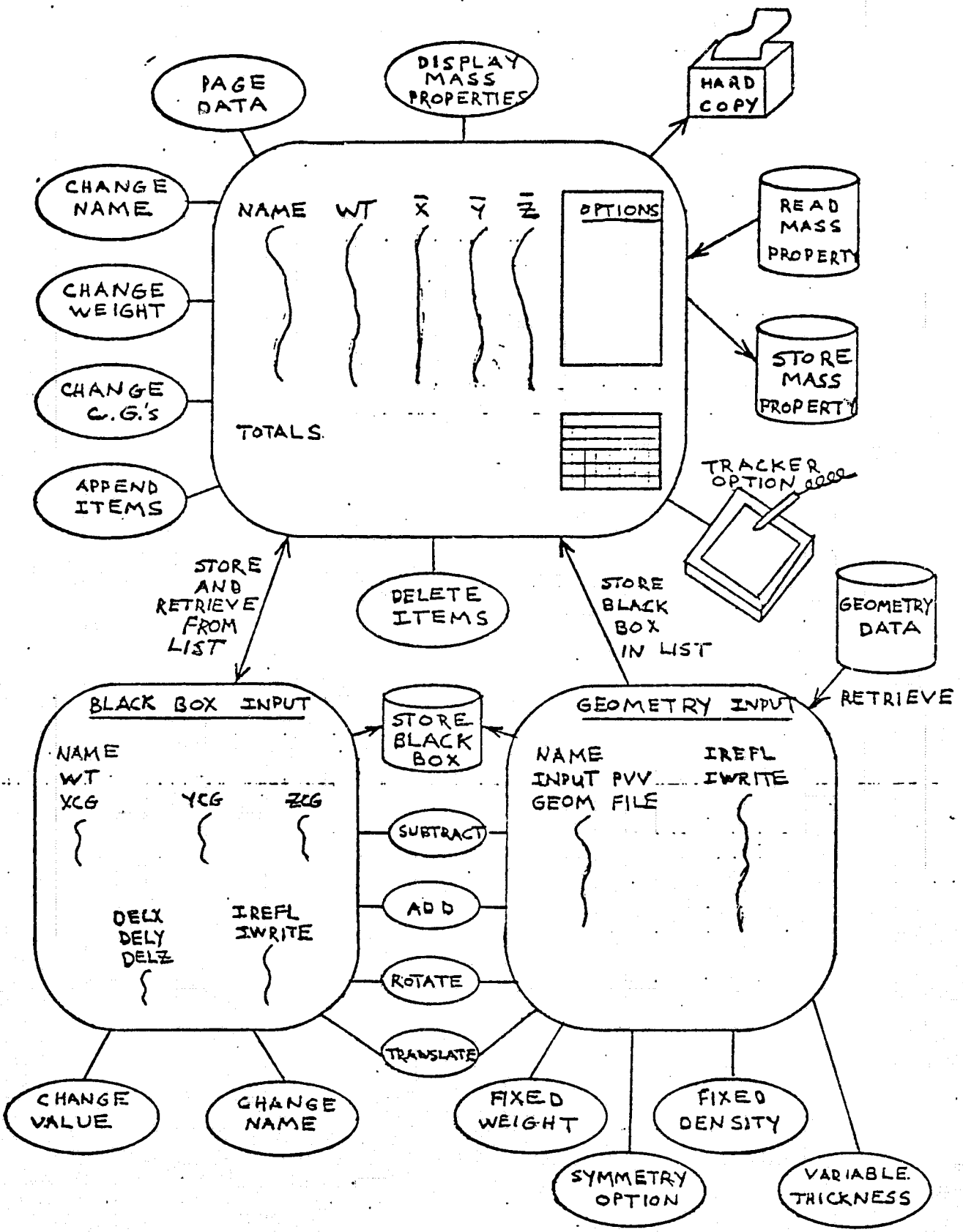


FIGURE 1 INTERACTIVE MASS PROPERTIES PROGRAM.

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.

VOLUME, AREA AND MASS PROPERTIES ANALYSIS

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 accumualted 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:

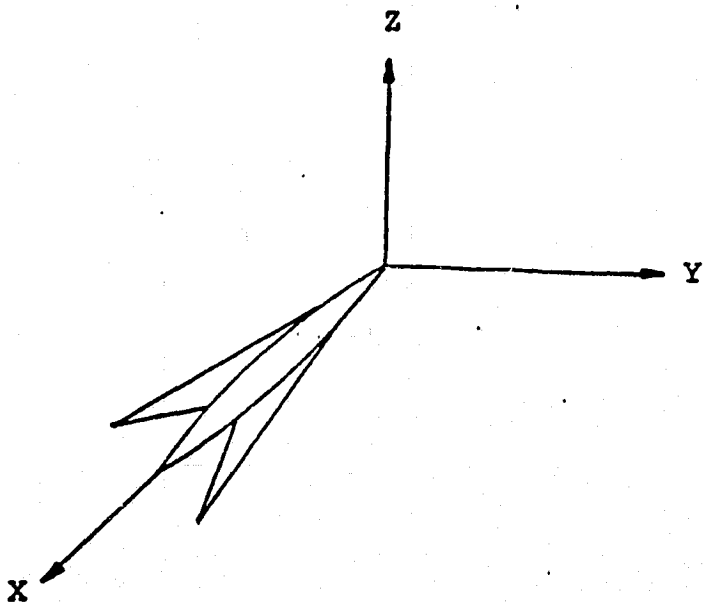
1. Weight.
2. Coordinates of the c.g.
3. Three mass moments of inertia plus three products of inertias.
4. Surface area.
5. Enclosed volume.
6. Projected area.

These properties are output for each segment along with cumulative totals and complete vehicle values.

Surface Model

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.

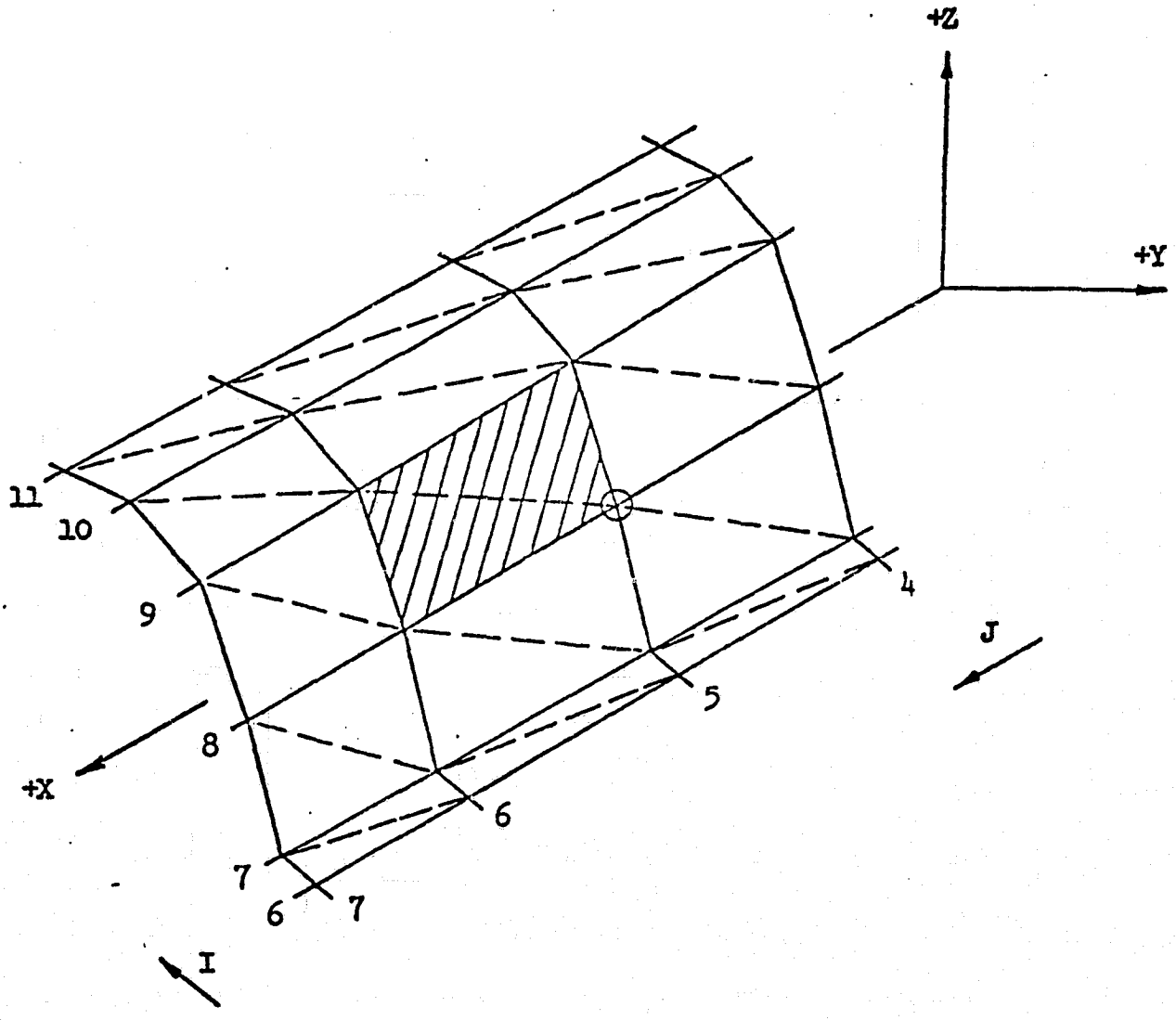
The configuration is usually (but not necessarily) positioned with its nose at the coordinate system origin and with the length of the body stretching in a positive X-direction. The coordinate system used for this program is the right hand Cartesian system as illustrated below:



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.



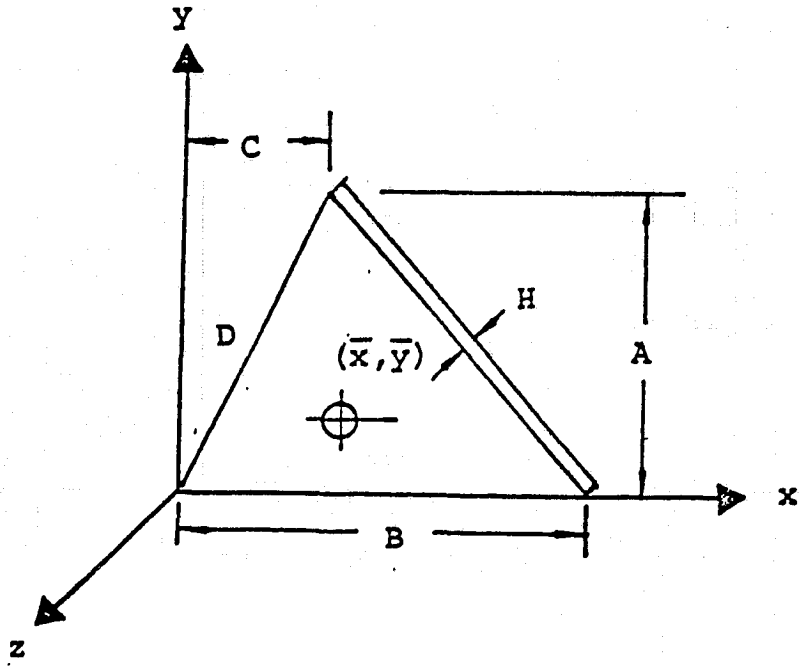
A portion of the mathematical surface model is presented in the illustration above. The X,Y,Z coordinates of the points where the solid grid lines cross are described to the program. Each point (I,J) is associated with a station number J and a contour point I. Thus, the circled point is 8,5 and it has coordinates $X_5; Y_{8,5}; Z_{8,5}$. 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. $\bar{A}$ 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:

$$S = AB/2 \tag{1}$$

The first moments of inertia for the same triangle are:

$$S_x = \rho HS(B+C)/Z \tag{2}$$

$$S_y = \rho HSA/2 \tag{3}$$

$$S_z = 0 \tag{4}$$

The second moments of inertia for the elemental triangle are:

$$I_{xx} = \rho HA^2S/6 \tag{5}$$

$$I_{yy} = \rho H(B^2+BC+C^2)S/6 \tag{6}$$

$$I_{zz} = I_{xx}+I_{yy} \tag{7}$$

Where ρ is the density of the thin triangle. Finally, the products of inertia are:

$$I_{xy} = \rho H(B+2C)AS/12 \tag{8}$$

$$I_{yz} = 0 \tag{9}$$

$$I_{xz} = 0 \tag{10}$$

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

$$\bar{x} = (B+C)/3 \tag{11}$$

$$\bar{y} = A/3 \tag{12}$$

The transformation of the inertias and products of inertia to the centroid yield:

$$\bar{I}_{xx} = SA^2/18 \qquad (13)$$

$$\bar{I}_{yy} = X(B^2-BC+C^2)/18 \qquad (14)$$

$$\bar{I}_{xy} = SA(ZC-B)/36 \qquad (15)$$

$$\bar{I}_{zz} = \bar{I}_{xx}+\bar{I}_{yy} \qquad (16)$$

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:

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

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:

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

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 vectices 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: $XO = X(J)$ (19)

$YO = Y(I,J)$ (20)

$ZO = Z(I,J)$ (21)

Then: $BX = 0.0$ (B lies in YZ Plane) (22)

$BY = Y(I+1,J) - YO$ (23)

$BZ = Z(I+1,J) - ZO$ (24)

$DX = X(J+1) - XO$ (25)

$DY = Y(I+1,J+1) - YO$ (26)

$DZ = Z(I-1,J-1) - ZO$ (27)

The cross product of B with D is a vector $\overline{S2}$ in the z direction. Its magnitude is twice the surface area of the triangle. Let SX,SY and SZ be the components of S2. Since BX = 0, the cross product yields:

$SX = BY*DZ - DY*BZ$ (28)

$SY = BZ*DX$ (29)

$SZ = -DX*BY$ (30)

Let B, D, S2 represent the magnitude of their respective vectors. Then $\overline{B}/B$ = unit vector in x direction and $\overline{S2}/S2$ = unit vector in z direction.

It follows:

$\overline{S2}/S2$ cross $\overline{B}/B$ = unit vector in y direction (31)

The components of each unit vector are a row in the matrix DC. Therefore:

$XPX = BX/B = 0.0$ (32)

$XPY = BY/B$ (33)

$XPZ = BZ/B$ (34)

$ZPX = SX/S2$ (35)

$ZPY = SY/S2$ (36)

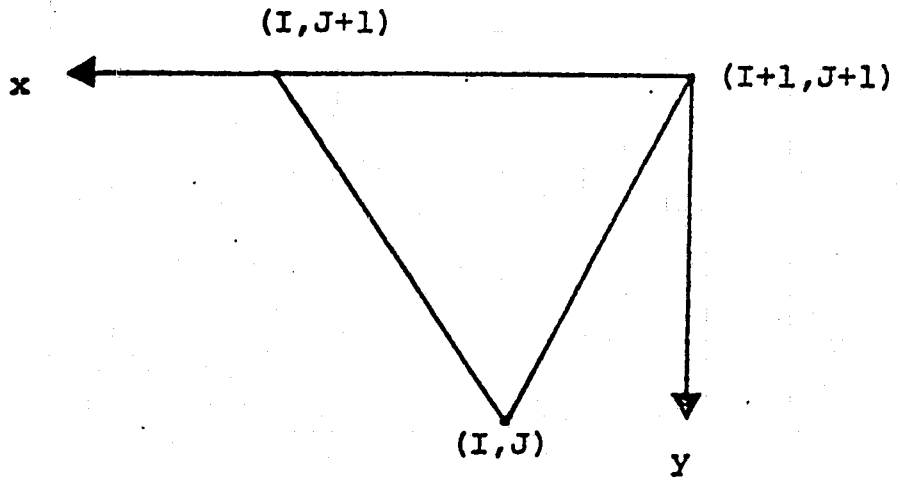
$ZPZ = SZ/S2$ (37)

$YPX = ZPX*XPZ - XPY*ZPZ = DX/A$ (38)

$$YPY = XPZ*ZPX \quad (39)$$

$$YPZ = ZPX*XPY \quad (40)$$

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,

$$XO = X(J+1) \quad (41)$$

$$YO = Y(I+1,J+1) \quad (42)$$

$$ZO = Z(I+1,J+1) \quad (43)$$

$$BX = 0.0 \quad (44)$$

$$BY = Y(I,J+1) - YO \quad (45)$$

$$BZ = Z(I,J+1) - ZO \quad (46)$$

$$DX = X(J) - XO \quad (47)$$

$$DY = Y(I,J) - YO \quad (48)$$

$$DZ = Z(I,J) - ZO \quad (49)$$

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.

$$C = DY*XPY+DZ*XPZ \tag{50}$$

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:

$$\begin{bmatrix} XBR \\ YBR \\ ZBR \end{bmatrix} = \begin{bmatrix} XO \\ YO \\ ZO \end{bmatrix} + [DC]^{-1} * \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix} \tag{51}$$

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:

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

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:

$$[PMI] = [DC]^T *[PMIP]*[DC] \tag{53}$$

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 S2 = twice the area of the triangle, S2 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:

$$A = 1/2 \; \Sigma \; S2_i \qquad (54)$$

S2 is twice the elemental area and the projected area (on the y-z plane) is:

$$A_p = 1/2 \sum_i SY_i \qquad (55)$$

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:

$$V = \sum_i \bar{x}_i \cdot ZPX_i \cdot A_i \qquad (56)$$

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

The center of volume is simply:

$$CV = \frac{1}{V} \; \Sigma \; \bar{X}^2_i \; ZPX_i \; A_i$$

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 ones 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.

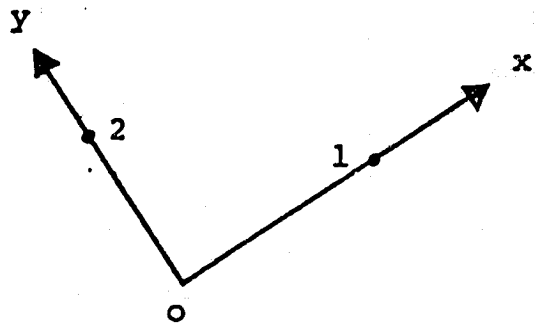
Black box masses are described to the program in the following manner:

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, $I_{XY} \neq I_{YZ} \neq 0.0$, 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:



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

1. Glatt, C. R. and Norton, P. J.: VAMP: A Computer Program for Calculating Volume, Area and Mass Properties of Aerospace Vehicles. NASA CR-2419, 1974.

2. Gregory, T. J., Peterson, R. J. and Wyss, J. A.: Performance Trade-Offs and Research Problems for Hypersonic Transports. AIAA Journal of Aircraft, July-August 1965.

3. Adams, J.D.: Vehicle Synthesis of High Speed Aircraft. VSAC, Volume I, USAF AFFDL-TR-71-40, 1971.

4. Oman, B.: Vehicle Synthesis for High Speed Aircraft. VSAC, Volume II, USAF AFFDL-TR-41-40, 1971.

5. Lee, V.A., Ball, H.G., Wadsworth, E.A., Moran, W.J. and McLead, J.D.: Computerized Aircraft Synthesis. AIAA Journal of Aircraft, September-October 1967.

6. Herbst, W.B. and Ross, H.: Application of Computer Aided Design Programs for the Management of Fighter Development Projects. AIAA Paper 70-364, presented at the AIAA Fighter Aircraft Concerence, March 1970.

7. Koenig, Robert W. and Fishback, Laurence H.: GENENG - A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines, NASA TN-D-6552, 1972.

8. Fishback, Laurence H. and Koenig, Robert W.: GENENG II - A Program for Calculating Design and Off-Design Performance for Two-and-Three Spool Turbofans with as Many as Three Nozzles, NASA TN-D-6553, 1972.

9. Giles, Gary L., Blackburn, Charles L. and Dixon, Sidney C.: Automated Procedure for Sizing Aerospace Vehicles Structures. Journal of Aircraft, Volume 9, No. 12, December 1972.

10. Sobieszczanski, J. and Soendorf, D.: A Mixed Optimization Method for Automated Design of Fuselage Structures. Journal of Aircraft, Volume 9, No. 12, December 1972, pp 805-811.

11. Giles, Gary L.: A Procedure for Automating Aircraft Wing Structural Design, Journal of Structures Division, American Society of Civil Engineers. January 1971, pp. 99-113.

12. Stroud, W. Jefferson, Dexter, Cornelia B. and Stein, Manual: Automated Preliminary Design of Simplified Wing Structures to Satisfy Strength and Flutter Requirements. NASA TN-D-6534, 1971.

13. Dwyer, Walter J., Emerton, Robert K. and Ojalvo, Irwing V.: An Automated Procedure for the Optimization of Practical Structures. Volume I, Theretical Development and User's Information. AFFDL-TR-70-118, USAF, April 1971.

14. Hague, D.S. and Glatt, C.R.: Optimal Design Integration of Military Flight Vehicles - ODIN/MFV. AFFDL-TR-73-132, 1973.

15. Glatt, C.R., Hague, D.S. and Watson, D.A.: DIALOG: An Executive Computer Program for Linking Independent Programs. NASA CR-2296, July 1973.

16. Craidon, C.B.: Description of a Digital Computer Program for Airplane Configuration Plots, NASA TM X-2074, September 1970.

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 existance 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

$PPCOPY 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.

```
1            SUBROUTINE AN6
2       C    A1.0
3       C    CONSTRUCTION OF ALPHA/NUMERIC INPUT DATA
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,RRNUM
7            DIMENSION TSUM(20),SN(20),ESUM(20),MASK(5)
10           DATA TSUM/20*0./,SUM/0./,LFLG(6)/0/,LFLG(3)/1/
11           DATA SN/20*0./,ESUM/20*0./
12           DATA MASK/0B,7700000000B,7777000000B,7777770000B,7777777700B/
13           IF(LFLG(3).EQ.0) GO TO 89
14      100  CALL ANRST
```

```
1     C   A1.1
2     C   ONE DATA ENTRY PER LIGHT PEN DEPRESS
3     C
4        89 IF(LFLG(6).EQ.1) GO TO 89
5     C   A1.2
6     C   WHEN CLR IS PICKED BY LIGHT PEN
7     C
10          IF(LFLG(9).EQ.1) GO TO 100
11    C   A1.3
12    C   CONSTRUCTION OF ALPHA WORD
13    C
14          IF(MFLAG(1).EQ.1) GO TO 60
15    C   A1.4
16    C   WHEN ENT IS PICKED BY LIGHT PEN
17    C
20          IF(LFLG(8).EQ.1) GO TO 999
21    C   A1.5
22    C   DETERMINE IF NUMERICAL INPUT IS REAL OR INTEGER
23    C
24          IF(LFLG(7)) 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
30       22 ICNT=ICNT+1
31          II=1-ICNT
32          L=K+II
33          IF(ICNT.EQ.1) GO TO 24
34          GO TO 30
35       24 LFLG(7)=0
36          GO TO 999
37       25 II=K-1
40       30 TSUM(K)=SUM
41          SN(K)=10.**II
42          LFLG(7)=0
43          GO TO 999
44       35 M=M+1
45          ESUM(M)=SUM
46          LFLG(7)=0
47          GO TO 999
50      999 EXP=INUM=0
51          RNUM=0.
52          IF(LFLG(5).NE.1) GO TO 43
53          DO 42 I=1,M
54          J=M-I
55          EXP=EXP+ESUM(I)*10.**(J)
56       42 CONTINUE
57          IF(LFLG(4).EQ.1) EXP=-EXP
60       43 DO 50 I=1,K
61          J=L+1-I
62          IF(I.GT.L) J=I
63          IF(LFLG(1).EQ.1) GO TO 45
64          J=K+1-I
65          INUM=INUM+TSUM(I)*SN(J)
66          GO TO 50
```

```
 67       45 RNUM=RNUM+TSUM(I)*SN(J)*10.**EXP
 70       50 CONTINUE
 71    C   A1.6
 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          IINUM=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(8).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          KCNT=ICNT-5
121       63 NDX=6*(KCNT-1)
122          NME(J)=(NME(J).AND.MASK(KCNT)).OR.(.NOT.MASK(KCNT).AND.
123         * (LETTER.R.NDX))
124          LFLG(7)=0
125          LFLG(2)=0
126          RETURN
```

```
1           SUBROUTINE ANRST
2     C     A2.0
3     C     RESET GRID INPUT VARIABLES
4     C
5           K=L=M=II=ICNT=IINUM=0
6           RRNUM=0.
7           NME(1)=NME(2)='      '
10          DO 60 I=1,10
11          LFLG(I)=0
12        60 CONTINUE
13          DO 70 I=1,20
14          ESUM(I)=0.
15          TSUM(I)=0.
16        70 CONTINUE
17          RETURN
20          END
```

```
1           SUBROUTINE AN7
2     C     N1.0
3     C     CONSTRUCTION OF ALPHA/NUMERIC INPUT FOR WABCOM
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,RRNUM
7           COMMON/LTR/ IFLAG(40)
10          DIMENSION TSUM(20),SN(20),ESUM(20),MASK(5)
11          LOGICAL IFLAG
12          DATA TSUM/20*0./,SUM/0./,LFLG(6)/0/,LFLG(3)/1/
13          DATA SN/20*0./,ESUM/20*0./,JCNT/0/
14          DATA MASK/0B,7700000000B,7777000000B,7777770000B,7777777700B/
15          JCNT=JCNT+1
16          IF(JCNT.EQ.1) GO TO 100
17          IF(LFLG(3).EQ.0) GO TO 89
20        100 CALL ANRST
```

```
1     C   N1.1
2     C   ONE DATA ENTRY PER LIGHT PEN PICK
3     C
4        89 IF(LFLG(6).EQ.1) GO TO 89
5     C   N1.2
6     C   WHEN CLR PICKED BY LIGHT PEN
7     C
10          IF(LFLG(9).EQ.1) GO TO 100
11    C   N1.3
12    C   CONSTRUCTION OF ALPHA WORD
13    C
14          IF(IFLAG(31)) GO TO 60
15    C   N1.4
16    C   WHEN ENT IS PICKED BY LIGHT PEN
17    C
20          IF(LFLG(8).EQ.1) GO TO 999
21    C   N1.5
22    C   DETERMINE IF NUMERICAL INPUT IS REAL OR INTEGER
23    C
24          IF(LFLG(7)) 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
30       22 ICNT=ICNT+1
31          II=1-ICNT
32          L=K+II
33          IF(ICNT.EQ.1) GO TO 24
34          GO TO 30
35       24 LFLG(7)=0
36          GO TO 999
37       25 II=K-1
40       30 TSUM(K)=SUM
41          SN(K)=10.**II
42          LFLG(7)=0
43          GO TO 999
44       35 M=M+1
45          ESUM(M)=SUM
46          LFLG(7)=0
47          GO TO 999
50      999 EXP=INUM=0
51          RNUM=0.
52          IF(LFLG(5).NE.1) GO TO 43
53          DO 42 I=1,M
54          J=M-I
55          EXP=EXP+ESUM(I)*10.**(J)
56       42 CONTINUE
57          IF(LFLG(4).EQ.1) EXP=-EXP
60       43 DO 50 I=1,K
61          J=L+1-I
62          IF(I.GT.L) J=I
63          IF(LFLG(1).EQ.1) GO TO 45
64          J=K+1-I
65          INUM=INUM+TSUM(I)*SN(J)
66          GO TO 50
```

```
 67        45 RNUM=RNUM+TSUM(I)*SN(J)*10.**EXP
 70        50 CONTINUE
 71     C   N1.6
 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           IINUM=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(8).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           KCNT=ICNT-5
121        63 NDX=6*(KCNT-1)
122           NME(J)=(NME(J).AND.MASK(KCNT)).OR.(.NOT.MASK(KCNT).AND.
123          * (LETTER.R.NDX))
124           LFLG(7)=0
125           LFLG(2)=0
126           RETURN
```

```
1           SUBROUTINE ANRST
2     C   N1.7
3     C   RESET GRID INPUT VARIABLES
4     C
5           K=L=M=II=ICNT=IINUM=0
6           RRNUM=0.
7           NME(1)=NME(2)='      '
10          DO 60 I=1,10
11          LFLG(I)=0
12       60 CONTINUE
13          DO 70 I=1,20
14          ESUM(I)=0.
15          TSUM(I)=0.
16       70 CONTINUE
17          RETURN
20          END
```

```
1           IMAGE GRIDW
2     C   R1.0
3     C   DISPLAY ALPHA/NUMERIC INPUT GRID
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,RRNUM
7           COMMON/GRD/ BBOX(50),TEST(200),TEST3(50),OPDPL(100),BOX(20)
10          INTEGER ALPHA(27),$PWD4
11          DATA ALPHA/1HA,1HG,1HM,1HS,1HB,1HH,1HN,1HT,1HC,1HI,1HO,1HU,
12         * 1HD,1HJ,1HP,1HV,1HE,1HK,1HQ,1HW,1HY,1HF,1HL,1HR,1HX,1HZ,1H//
13          DATA LFLG(10)/0/
14          LDX(1.F)
15          LDY(-1.F)
16          LSCL(.4F)
17          TABLE2D(BBOX)
20          CHAR(TEST3)
21          IF(LFLG(6).EQ.0) GO TO 200
22          IF(($LPNSW.AND.200B).NE.0) GO TO 200
23          LFLG(6)=0
24        200 CLEARPEN(1)
25          $PWD4=0
26          PENON(1)
27          MENU("0B0X0061@Y0070A0.0A0X0066@Y0070G0.0A0
30         * 0X0073@Y0070M0.0A0X0077@Y0070S0.0A0
31         * 0X0061@Y0074B0.0A0X0066@Y0074H0.0A0
32         * 0X0073@Y0074N0.0A0X0077@Y0074T0.0A0
33         * 0X0061@Y0100C0.0A0X0066@Y0100I0.0A0
34         * 0X0073@Y0100O0.0A0X0077@Y0100U0.0A0
35         * 0X0061@Y0106D0.0A0X0066@Y0106J0.0A0
36         * 0X0073@Y0106P0.0A0X0077@Y0106V0.0A0
37         * 0X0061@Y0112E0.0A0X0066@Y0112K0.0A0
40         * 0X0073@Y0112Q0.0A0X0077@Y0112W0.0A0
41         * 0X0104@Y0112Y0.0A0X0061@Y0120F0.0A0
42         * 0X0066@Y0120L0.0A0X0073@Y0120R0.0A0
43         * 0X0077@Y0120X0.0A0X0104@Y0120Z0.0A0
44         * 0X0111@Y0120/0.0A0
45         * 0X0104@Y0070I0.0A0X0111@Y0070Z0.0A0
46         * 0X0115@Y0070J0.0A0X0104@Y0074A0.0A0
47         * 0X0111@Y0074S0.0A0X0115@Y0074G0.0A0
50         * 0X0104@Y0100Y0.0A0X0111@Y0100B0.0A0
51         * 0X0115@Y0100P0.0A0X0104@Y0106O0.0A0
52         * 0X0111@Y0106*0.0A0S0X0115@Y0106CLR0.0A0
53         * 0E0X0111@Y0112-0.0A0S0X0116@Y0112EX0.0A0
54         * 0S0X0115@Y0120ENT0.0Z")
55          PENOFF(1)
56          IF(PENHIT(1)) LFLG(10)=1
57          GO TO (16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,
60         * 16,16,16,16,16,16,16,16,16,16,16,
61         * 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
62          RETURN
63        1 IF(LFLG(6).EQ.1) RETURN
64          SUM=1.
65          LETTER=1H1
66          LFLG(6)=1
```

```
 67          LFLG(7)=1
 70          $PWD4=0
 71          RETURN
 72        2 IF(LFLG(6).EQ.1) RETURN
 73          SUM=2.
 74          LETTER=1H2
 75          LFLG(6)=1
 76          LFLG(7)=1
 77          $PWD4=0
100          RETURN
101        3 IF(LFLG(6).EQ.1) RETURN
102          SUM=3.
103          LETTER=1H3
104          LFLG(6)=1
105          LFLG(7)=1
106          $PWD4=0
107          RETURN
110        4 IF(LFLG(6).EQ.1) RETURN
111          SUM=4.
112          LETTER=1H4
113          LFLG(6)=1
114          LFLG(7)=1
115          $PWD4=0
116          RETURN
117        5 IF(LFLG(6).EQ.1) RETURN
120          SUM=5.
121          LETTER=1H5
122          LFLG(6)=1
123          LFLG(7)=1
124          $PWD4=0
125          RETURN
126        6 IF(LFLG(6).EQ.1) RETURN
127          SUM=6.
130          LETTER=1H6
131          LFLG(6)=1
132          LFLG(7)=1
133          $PWD4=0
134          RETURN
135        7 IF(LFLG(6).EQ.1) RETURN
136          SUM=7.
137          LETTER=1H7
140          LFLG(6)=1
141          LFLG(7)=1
142          $PWD4=0
143          RETURN
144        8 IF(LFLG(6).EQ.1) RETURN
145          SUM=8.
146          LETTER=1H8
147          LFLG(6)=1
150          LFLG(7)=1
151          $PWD4=0
152          RETURN
153        9 IF(LFLG(6).EQ.1) RETURN
154          SUM=9.
```

```
155          LETTER=1H9
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=1H0
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=1H*
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=1H-
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
```

```
243          END
```

```
1           SUBROUTINE BLDNAM(NAM)
2     C   J1.0
3     C   FORMATS FILE INPUT NAME FOR FILE ACCESSING SUBROUTINES
4     C
5           IMPLICIT INTEGER (A-Z)
6           DIMENSION NAM(3)
7           DATA MASK/77B/
10    C     IF(NAM(2).EQ.1H ) NAM(2)=NAM(1)
11    C     IF(NAM(1).EQ.NAM(2)) RETURN
12          NAM(3)=NAM(2)
13          NAM(2)=NAM(1)
14          DO 10 I=9,0,-1
15          WD=I/5+2
16          KNT=(6*I)//30
17          ISHFT=24-KNT
20          TM=MASK.L.ISHFT
21          POSC=NAM(WD).AND.TM
22          CHAR=(POSC.L.KNT).OR.1H[
23          IF((CHAR.NE.1H[).AND.(CHAR.NE.1H )) GOTO 20
24          NAM(WD)=NAM(WD).AND.(-TM)
25    10    CONTINUE
26    20    CONTINUE
27          NAM(1)=(@NAM+1).AND.77777B
30          NAM(1)=NAM(1).OR.7777500000B
31          RETURN
32          END
```

```
1           SUBROUTINE BLDNME(NAME)
2     C   B1.0
3     C   FORMATS FILE INPUT NAME FOR FILE ACCESSING SUBROUTINES
4     C
5           DIMENSION L(10),NAME(3)
6           NINE=9
7           READ(NINE,100) L
10    100   FORMAT(10R1)
11          DO 3 M=1,3
12    3     NAME(M)=0
13          J=2
14          NL=0
15          NBLNK=0
16          DO 50 I=1,10
17          IF(L(I).EQ.40B) GO TO 15
20          IF(NBLNK.NE.0) GO TO 20
21    5     IDUM=L(I)
22          CALL IT
23          GO TO 50
24    15    NBLNK=NBLNK+1
25          GO TO 50
26    20    IDUM=40B
27          DO 21 L=1,NBLNK
30    21    CALL IT
31          NBLNK=0
32          GO TO 5
33    50    CONTINUE
34          IF(NBLNK.EQ.0) GO TO 60
35          IDUM=0
36          DO 70 L=1,NBLNK
37    70    CALL IT
40          NBLNK=0
41    60    IF(NAME(3).NE.0) GO TO 80
42          NAME(1)=NAME(2)
43          RETURN
44    80    LOC=0NAME(2)
45          NAME(1)=(.H.7777SB) .O. LOC
46          RETURN
47    C
50          SUBROUTINE IT
51          NAME(J)=.K.NAME(J)
52          NAME(J)=NAME(J).O.IDUM
53          NL=NL+1
54          IF(NL.GE.5) J=3
55          RETURN
56          END
```

```
1           SUBROUTINE CGMOVE
2     C     C1.0
3     C     BLACK BOX TRACKER CG TRANSLATOR
4     C
5           COMMON/MOVIT/NAME(2),XCG,YCG,ZCG,NREAD
6           COMMON/RSTORE/CAMB,ALEN,RADZ(30),RADY(30),RADX(30),CAMX(30),
7          * CAMZ(30),FACTOR,IBLOCK,FILLER(56)
10          LOGICAL FLAG,ADTF1,ADTF2,TOP,NREAD
11          LOGICAL FPEN,FPENP,NDSTP
12          FRACTION RADX,RADZ,SX,SY,RADTX,RADTY,POSTX,POSTY,POSTZ
13          FRACTION RX,RY,RADY,BIAS,FDUM(30)
14          DIMENSION NAMDUM(3),IDUM(30),NOTES(50),NOTES1(50),NOTES2(50)
15          DIMENSION TVIEW(100),SVIEW(100),DUMPIT(210)
16          EQUIVALENCE(DUMPIT,CAMB)
17          EQUIVALENCE(NPTS,RADZ(1)),(NPTSC,CAMZ(1)),(NPTSY,RADY(1))
20    C
21    C
22          $NDSTP=.TRUE.
23    C     C1.1
24    C     DINGER DATA
25    C
26          DATA2D(DINGER)
27          ZSET(1.F)
30          MOVE(1.F,1.F)
31          DRAW(1.F,-1.F)
32          DRAW(-1.F,-1.F)
33          DRAW(-1.F,1.F)
34          DRAW(1.F,1.F)
35          ENDLIST
36          ENDDATA
37    C
40    C
41    C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
42    C
43    1       IF(NDSTP) GO TO 1
44            IF(NREAD) GO TO 2000
45    C     C1.2
46    C     INITIALIZE CG POSITION
47    C
50    72      POSTX=-RTOF(XCG/FACTOR+.8)
51            POSTY=RTOF(YCG/FACTOR)
52            POSTZ=RTOF(ZCG/FACTOR)
53    C     C1.3
54    C     INITIALIZE LABELS
55    C
56            LABEL(NOTES)
57            WRITE(16,40) NAME
60    40      FORMAT("0B0X0030"Y0014COMPONENT0X0074XCG0X0115YCG0X0135ZCG0X0300Y
61           *0020",2A5,"0.")
62            ENDLIST
63            LABEL(NOTES1)
64            WRITE(16,41)XCG,ZCG
65    41      FORMAT("0B0X0065"Y0020",F7.1,"0X0125",F7.1,"0.")
66            ENDLIST
```

```
 67          LABEL(NOTES2)
 70          WRITE(16,42)YCG
 71    42    FORMAT("@B@X@105@Y@020",F7.1,"@.")
 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("@S@B@X@067@Y@150X=",F6.0,"  Y=",F6.0)
125          GO TO 37
126    36    WRITE(16,91) ENGX,ENGZ
127    91    FORMAT("@S@B@X@067@Y@150X=",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   C1.8
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..')
10          READ(NINE,0) LPV
11          LLPV=IOCT(LPV)
12    2015  WRITE(NINE,2002)
13    2002  FORMAT(' ENTER NAME OF INPUT FILE..')
14          CALL BLDNME(NAMDUM)
15          $NEWID=5000000000B
16          $DFILETYPE=17B
17          IFILAD=1
20          CALL RDSKOK(LLPV,NAMDUM,208,IFILAD,IFLG)
21          IF(IFLG.NE.0) GO TO 2010
22          WRITE(10,2003)NAMDUM(2),NAMDUM(3),LLPV
23    2003  FORMAT(' CAN NOT LOCATE ',2A5,' ON PACK/VOL ',O4)
24          GO TO 2010
25    2010  CALL RNDRD(DUMPIT,0,208,IFILAD,IOFF)
26          CALL DSKER
27          NEWID=7000000000B
30          DFILETYPE=7B
31          CALL SKETCH
32          NREAD=.FALSE.
33          GO TO 72
34    C
```

```
C
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      SUBROUTINE SKETCH
C   C1.9
C   CREATE FUSELAGE OUTLINE
C
      ISS=NPTS+1
      DO 2 IS=2,ISS
      FDUM(IS)=-RADZ(IS)
2     IF(CAMB) FDUM(IS)=FDUM(IS)+CAMZ(IS)+CAMZ(IS)
      LIST2D(SVIEW)
      ZSET(0.F)
      DO 3 IS=2,ISS
      IF(IS.GT.2) GO TO 3
      MOVE(RADX(IS),RADZ(IS))
3     DRAW(RADX(IS),RADZ(IS))
      DO 4 IS=2,ISS
      IF(IS.GT.2) GO TO 4
      MOVE(RADX(IS),FDUM(IS))
4     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))
5     DRAW(RADX(IS),RADY(IS))
      DO 6 IS=2,ISS
      IF(IS.GT.2) GO TO 6
      MOVE(RADX(IS),-RADY(IS))
6     DRAW(RADX(IS),-RADY(IS))
      ENDLIST
      RETURN
```

```
C
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
        IMAGE TWERP
C     C2.0
C     DISPLAY FUSELAGE
C
        READ(3)
C     C2.1
C     CHECK DDT PEN FOR TIP SWITCH
C
        FPEN=.FALSE.
        IF(.N.$ADTF1 .A. $ADTF2) FPEN=.TRUE.
        IF(FPEN.A..N.FPENP) FLAG=.TRUE.
        IF(.N. FPEN) GO TO 5
C     C2.2
C     SAVE DDT PEN POSITION
C
        SX=$RADTX
        SY=$RADTY
5       FPENP=FPEN
C     C2.3
C     GRID DATA
C
        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
C     SHOW SIDEVIEW & GRID
C
        TABLE2D(SVIEW)
        TABLE2D(GRD)
        POSCHAR(-.82F,.42F,0.F."@S@B-Z")
        POSCHAR(.83F,-.01F,0.F,"@S@B-X")
        POSCHAR(-.2F,.45F,0.F,"@BSIDEVIEW")
C     C2.5
C     POSITION & DISPLAY SIDEVIEW CG
C
        DY(POSTZ)
        DX(POSTX)
        LSCL(.1F)
        TABLE2D(DINGER)
C     C2.6
C     SHOW TOPVIEW & GRID
C
        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,"&B&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)
123          TABLE2D(DINGER)
124          RETURN
125          END
```

```
1           SUBROUTINE CROSS(A,B,C,CM)
2     C
3     C     UNITIZED MATRIX CROSS PRODUCT    C = A X B
4     C
5           DIMENSION A(3),B(3),C(3)
6           C(1)=A(2)*B(3)-A(3)*B(2)
7           C(2)=A(3)*B(1)-A(1)*B(3)
10          C(3)=A(1)*B(2)-A(2)*B(1)
11          CM=SQRT(C(1)*C(1)+C(2)*C(2)+C(3)*C(3))
12          RETURN
13          END
```

```
1           SUBROUTINE DEC FMT (VALUE,NFIELD,FMT)
2      C    D1.0
3      C    BUILD FORMAT FOR DECIMAL NUMBERS
4      C
5           INTEGER FMT,CVT(2),POINT
6           DATA (CVT(I),I=1,2) /1HF,1HE/
7           DATA POINT/1H./
10     C * * . . . . . . . . . . . . . . . DETERMINE THE EXPONENT (NPWR
11          NPWR=0
12          ABVAL=ABS(( 1.+10.**(3-NFIELD))*VALUE)
13          IF (ABVAL.LT.1.) GO TO 550
14          DO 530 I=1,99
15          ABVAL=ABVAL/10.
16          IF (ABVAL.LT.1.) GO TO 540
17        530 CONTINUE
20        540 CONTINUE
21          NPWR=I
22          GO TO 600
23        550 CONTINUE
24          DO 570 I=1,99
25          ABVAL=ABVAL*10.
26          IF (ABVAL.GT..1) GO TO 580
27        570 CONTINUE
30        580 CONTINUE
31          NPWR=-I
32     C * * . . . . . . . . . . . . . . . SELECT THE FORMAT
33        600 CONTINUE
34          NSP=1
35          IF (VALUE.LT.0) NSP=2
36          IF (NPWR.GE.0) GO TO 620
37          IF (NPWR.LT.-4) GO TO 640
40          NDEC=NFIELD-NSP
41          IFMT=1
42          GO TO 690
43        620 CONTINUE
44          NDEC=NFIELD-NPWR-NSP
45          IF (NDEC.LE.0) GO TO 640
46          IFMT=1
47          GO TO 690
50        640 CONTINUE
51          NDEC=NFIELD-NSP-5
52          IFMT=2
53        690 CONTINUE
54          IF (NDEC.LT.0) NDEC=0
55     C * * . . . . . . . . . . . . . . . BUILD THE FORMAT
56          ENCODE(5,1000,FMT) CVT(IFMT),NFIELD,POINT,NDEC
57       1000 FORMAT (A1,I2,A1,I1)
60          RETURN
61          END
```

```
1           IMAGE GRID
2     C     G1.0
3     C     DISPLAY ALPHA/NUMERIC INPUT GRID
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,RRNUM
7           COMMON/GRD/ BBOX(50),TEST,TEST3
10          INTEGER ALPHA(27),$PWD4
11          DATA ALPHA/1HA,1HG,1HM,1HS,1HB,1HH,1HN,1HT,1HC,1HI,1HO,1HU,
12         * 1HD,1HJ,1HP,1HV,1HE,1HK,1HQ,1HW,1HY,1HF,1HL,1HR,1HX,1HZ,1H//
13          DATA LFLG(10)/0/
14          LDX(1.F)
15          LDY(-1.F)
16          LSCL(.4F)
17          TABLE2D(BBOX)
20          CHAR(TEST)
21          IF(LFLG(6).EQ.0) GO TO 200
22          IF((SLPNSW.AND.200B).NE.0) GO TO 200
23          LFLG(6)=0
24    200   CLEARPEN(1)
25          $PWD4=0
26          PENON(1)
27          MENU("0B0X0061@Y0070A0.0A0X0066@Y0070G0.0A0
30         * 0X0073@Y0070M0.0A0X0077@Y0070S0.0A0
31         * 0X0061@Y0074B0.0A0X0066@Y0074H0.0A0
32         * 0X0073@Y0074N0.0A0X0077@Y0074T0.0A0
33         * 0X0061@Y0100C0.0A0X0066@Y0100I0.0A0
34         * 0X0073@Y0100O0.0A0X0077@Y0100U0.0A0
35         * 0X0061@Y0106D0.0A0X0066@Y0106J0.0A0
36         * 0X0073@Y0106P0.0A0X0077@Y0106V0.0A0
37         * 0X0061@Y0112E0.0A0X0066@Y0112K0.0A0
40         * 0X0073@Y0112Q0.0A0X0077@Y0112W0.0A0
41         * 0X0104@Y0112Y0.0A0X0061@Y0120F0.0A0
42         * 0X0066@Y0120L0.0A0X0073@Y0120R0.0A0
43         * 0X0077@Y0120X0.0A0X0104@Y0120Z0.0A0
44         * 0X0111@Y0120/0.0A0
45         * 0X0104@Y0070100.0A0X0111@Y0070200.0A0
46         * 0X0115@Y0070300.0A0X0104@Y0074400.0A0
47         * 0X0111@Y0074500.0A0X0115@Y0074600.0A0
50         * 0X0104@Y0100700.0A0X0111@Y0100800.0A0
51         * 0X0115@Y0100900.0A0X0104@Y0106000.0A0
52         * 0X0111@Y0106*0.0A0S0X0115@Y0106CLR0.0A0
53         * 0E0X0111@Y0112-0.0A0S0X0116@Y0112EX0.0A0
54         * 0S0X0115@Y0120ENT0.0Z")
55          PENOFF(1)
56          IF(PENHIT(1)) LFLG(10)=1
57          GO TO (16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,
60         * 16,16,16,16,16,16,16,16,16,16,16,
61         * 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
62          RETURN
63        1 IF(LFLG(6).EQ.1) RETURN
64          SUM=1.
65          LETTER=1H1
66          LFLG(6)=1
```

```
 67          LFLG(7)=1
 70          $PWD4=0
 71          RETURN
 72        2 IF(LFLG(6).EQ.1) RETURN
 73          SUM=2.
 74          LETTER=1H2
 75          LFLG(6)=1
 76          LFLG(7)=1
 77          $PWD4=0
100          RETURN
101        3 IF(LFLG(6).EQ.1) RETURN
102          SUM=3.
103          LETTER=1H3
104          LFLG(6)=1
105          LFLG(7)=1
106          $PWD4=0
107          RETURN
110        4 IF(LFLG(6).EQ.1) RETURN
111          SUM=4.
112          LETTER=1H4
113          LFLG(6)=1
114          LFLG(7)=1
115          $PWD4=0
116          RETURN
117        5 IF(LFLG(6).EQ.1) RETURN
120          SUM=5.
121          LETTER=1H5
122          LFLG(6)=1
123          LFLG(7)=1
124          $PWD4=0
125          RETURN
126        6 IF(LFLG(6).EQ.1) RETURN
127          SUM=6.
130          LETTER=1H6
131          LFLG(6)=1
132          LFLG(7)=1
133          $PWD4=0
134          RETURN
135        7 IF(LFLG(6).EQ.1) RETURN
136          SUM=7.
137          LETTER=1H7
140          LFLG(6)=1
141          LFLG(7)=1
142          $PWD4=0
143          RETURN
144        8 IF(LFLG(6).EQ.1) RETURN
145          SUM=8.
146          LETTER=1H8
147          LFLG(6)=1
150          LFLG(7)=1
151          $PWD4=0
152          RETURN
153        9 IF(LFLG(6).EQ.1) RETURN
154          SUM=9.
```

```
155          LETTER=1H9
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=1H0
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=1H*
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=1H-
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
```

```
243          END
```

```
1           FUNCTION IOCT(NN)
2     C     U1.0
3     C     COMPUTES OCTAL EQUIVALENCE FROM DECIMAL INPUT
4     C
5           ITMP=0
6           ITMP=((NN/1000)*512)+ITMP
7           NRMD=NN//1000
10          NTMP=((NRMD/100)*64)+ITMP
11          ITMP=NRMD//100
12          JTMP=((ITMP/10)*8)+NTMP
13          IOCT=(ITMP//10)+JTMP
14          RETURN
15          END
```

```
1           IMAGE OPTIONS
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,RRNUM
7           COMMON/OPT/ NHIT,OPDPL
10          INTEGER $PWD4
11          DATA NHIT/0/
12          DATA2D (BOX)
13          ZSET(0.F)
14          MOVE(.4F,.4F)
15          DRAW(.4F,-1.F)
16          DRAW(-.4F,-1.F)
17          DRAW(-.4F,.4F)
20          DRAW(.4F,.4F)
21          ENDLIST
22          ENDDATA
23          LDX(1.F)
24          LDY(1.F)
25          TABLE2D (BOX)
26          CHAR("0B0X0071@Y0064OPTIONS")
27          CHAR(OPDPL)
30          IF(NHIT.EQ.0) GO TO 100
31          IF((SLPNSW.AND.200B).NE.0) GO TO 100
32          NHIT=0
33       100 CLEARPEN(1)
34          $PWD4=0
35          PENON(1)
36          MENU("0B0S0X0061@Y0072PACK/VOL@.0A0
37         * 0X0061@Y0076SELECT FILE@.0A0
40         * 0X0061@Y0102LIST VOL@.0A0
41         * 0X0061@Y0106BLACK BOX@.0A0
42         * 0X0061@Y0112GEOMETRY@.0A0
43         * 0X0061@Y0116DISPLAY@.0A0
44         * 0X0061@Y0122APPEND DISPLAY@.0A0
45         * 0X0061@Y0126DELETE COMPONENT@.0A0
46         * 0X0061@Y0132STORE@.0A0
47         * 0X0061@Y0136INPUT NEW FILE@.0A0
50         * 0X0061@Y0142BLACK BOX TRACKER@.0A0
51         * 0X0061@Y0146PRINT@.0Z")
52          PENOFF(1)
53          IF((LF_G(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1))GOTO 200
54          IF($PWD4.EQ.0) GO TO 200
55          IF(NHIT.EQ.1) GO TO 200
56          NFLAG($PWD4)=1
57          MFLAG(3)=1
60          $PWD4=0
61       200 CALLSIMG GRID
62          RETURN
63          END
```

```
1     F      INCLUDE=FORMT
2     C    F1.0
3     C    FILE LOCATOR BY FILE SEQ. NUMBER IN PACK/VOL
4     C
5            SUBROUTINE LOCFLE (IPV,NAMER,IFILNO)
6            COMMON/PARA/ LFLG(10),MFLAG(4),SUM,LETTER,NO,NFLG,INO,NFLAG(12)
7           * ,JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),IINUM,RRNUM
10           COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
11          * VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
12          * VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
13           REAL VALUE1,VALUE2,VALUE3,VALUE4,VALUE5,
14          * VALUE6,VALUE7,VALUE8,VALUE9,VALUE10,
15          * VALUE11,VALUE12,VALUE13
16           INTEGER  NAME,NAMER(3)
17           CALL BLDNAM(NAMER)
20           CALL DSVOL(IPV,IVFLAG,NFREES)
21           CALL DSKER
22    C      WRITE(10,100) IVFLAG,NFREES
23    C   100 FORMAT(1X,5013)
24           CALL DSFIL(IPV,NAMER,IFFLAG,NMFIL,IFILAD,IFILNO)
25           CALL DSKER
26    C      WRITE(10,100)IFFLAG,NMFIL,IFILAD,IFILNO
27           RETURN
30           END
```

```
1     F      INCLUDE=FORMT
2            SUBROUTINE INPUT(IPV)
3     C    I1.0
4     C    INPUT NEW DATA FILE
5     C
6            COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
7           * VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
10          * VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
11           DIMENSION IBUF(1000),DUMN(2)
12           REAL VALUE1,VALUE2,VALUE3,VALUE4,VALUE5,
13          * VALUE6,VALUE7,VALUE8,VALUE9,VALUE10,
14          * VALUE11,VALUE12,VALUE13
15           INTEGER NAME(2,60),NAMER(3)
16           DATA DUMN(1)/5H      /,DUMN(2)/5H      /,DUMX/0./,DUMY/0./,DUMZ/0./
17           DATA NINE/9/
20           DATA FMT2/1515151515B/,FMT3/7777777777B/
21           DATA FMT4/7777700000B/
22           NTOT=0
23           CALL PTCH9
24           WRITE(NINE,3)
25         3 FORMAT(//,'     ENTER NAME OF NEW FILE...')
26           READ(NINE,5) NAMER
27         5 FORMAT(2A5)
30           CALL BLDNAM(NAMER)
31        10 WRITE(NINE,1776)
32      1776 FORMAT(//,' ENTER NAME OF COMPONENT...')
33           NTOT=NTOT+1
34           READ(NINE,100) NAME(1,NTOT),NAME(2,NTOT)
35       100 FORMAT(2A5)
36           IF(NAME(1,NTOT).EQ.4HDONE) GO TO 25
37           WRITE(NINE,1976)
40      1976 FORMAT(/,' ENTER 13 VALUES...')
41           READ(NINE,0) VALUE1(NTOT),VALUE2(NTOT),VALUE3(NTOT)
42          * ,VALUE4(NTOT),VALUE5(NTOT),VALUE6(NTOT),VALUE7(NTOT)
43          * ,VALUE8(NTOT),VALUE9(NTOT),VALUE10(NTOT),VALUE11(NTOT)
44          * ,VALUE12(NTOT),VALUE13(NTOT)
45           GO TO 10
46        25 NTOT=NTOT-1
47           ENCODE(5,1000,IBUF(1)) NTOT,FMT2
50      1000 FORMAT(I3,A2)
51           J=2
52           DO 5000 I=1,NTOT
53           ENCODE(75,2000,IBUF(J)) NAME(1,I),NAME(2,I),VALUE1(I),VALUE2(I),
54          * VALUE3(I),VALUE4(I),VALUE5(I),VALUE6(I),FMT2
55      2000 FORMAT(2A5,6F10.1,A5)
56           ENCODE(75,3000,IBUF(J+15))VALUE7(I),VALUE8(I),VALUE9(I),
57          * VALUE10(I),VALUE11(I),VALUE12(I),VALUE13(I),FMT2
60      3000 FORMAT(7F10.1,A5)
61           ENCODE(45,7000,IBUF(J+30)) DUMN(1),DUMN(2),DUMX,DUMY,DUMZ,FMT2
62      7000 FORMAT(2A5,3F10.1,A5)
63           J=J+39
64      5000 CONTINUE
65           ENCODE(5,4000,IBUF(J))FMT4
66      4000 FORMAT(A5)
```

```
67          ENCODE(5,6000,IBUF(J+1))FMT3
70     6000 FORMAT(A5)
71          LENGTH=J
72          CALL STRTF(IPV,NAMER)
73          CALL APNDF(IBUF(1),LENGTH)
74          CALL APNDF(IBUF(J+1),1)
75          CALL CLOSF
76          RETURN
77          END
```

```
1           SUBROUTINE LOAD(A,T,N)
2           DIMENSION T(1),A(1)
3           DO 1 I=1,N
4           T(I)=A(I)
5         1 CONTINUE
6           RETURN
7           END
```

```
      PROGRAM MAIN
C     M1.0
C     MAIN CONTROL FOR IMP
C
      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
      COMMON/GRD/ BBOX(50),TEST,TEST3
      COMMON/CNT/ IPASS,NUP,NAMER(3),IPV,ISTRT,IEND,IT,LINES
      COMMON/NMES/ NMBB(2)
      COMMON/MOVIT/ NAMDT(2),XCG,YCG,ZCG,IDTFLG
      COMMON/OPT/ NHIT,OPDPL
      COMMON/WB/ NMB(2),WT(15),TEMP(18),IWRITE,IJ,ITNM(2),TIXX(15),NUMO
     * ,IPRINT,PWT(3),DCG(60,3),IGNME(3,60)
      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,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,NAMER,NME,IPARA(5),NMDSPL(2),IAPN(5)
      LOGICAL IDTFLG
      DATA IINUM/0/,RRNUM/0/,NME/2*'      '/,MFLAG/4*0/,IDTFLG/.TRUE./
      DATA IPARA/1H ,4H WT ,4HX-CG,4HY-CG,4HZ-CG/,IDPV/110/,IPV/110B/
      DATA IAPN/5H NAME,5H   WT,5H X-CG,5H Y-CG,5H Z-CG/
      DATA NFLAG/12*0/,LFLG(10)/0/,LINES/20/,NMRDPL(1)/5HDATA /
      DATA NMRDPL(2)/5H     /,PWT/3*0./
      DATA2D (BBOX)
      ZSET(0.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(.7142F,-1.F)
```

```
1           DRAW(.7142F,.7142F)
2           MOVE(.4285F,.7142F)
3           DRAW(.4285F,-1.F)
4           MOVE(.1428F,-1.F)
5           DRAW(.1428F,.7142F)
6           MOVE(-.1428F,.7142F)
7           DRAW(-.1428F,-1.F)
10          MOVE(-.4285F,-1.F)
11          DRAW(-.4285F,.7142F)
12          MOVE(-.7142F,.7142F)
13          DRAW(-.7142F,-1.F)
14          ENDLIST
15          ENDDATA
16    C
17          CLEAR(OPDPL)
20       30 DISPLAY OPTIONS
21          DO 39 I=1,10000
22       39 IX2=I
23    C     M1.1
24    C     INITIALIZE PROGRAM VARIABLES
25    C
26       40 CALL RESET2
27       50 CALL RESET1
30      100 LABEL (TEST)
31          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)
34      120 FORMAT("@B@X@060@Y@056",2A5,2X,A5)
35      130 IF(JJJ.NE.0) GO TO 170
36          IF(LFLG(1).EQ.1) GO TO 150
37          WRITE(16,140) IINUM
40      140 FORMAT("@B@X@070@Y@064",I10)
41          GO TO 190
42      150 WRITE(16,160) RRNUM
43      160 FORMAT("@B@X@070@Y@064",F10.5)
44          GO TO 190
45      170 WRITE(16,180) NME(1),NME(2)
46      180 FORMAT("@B@X@070@Y@064",2A5)
47      190 ENDLIST
50          LABEL(OPDPL)
51          ZSET(0.0F)
52          WRITE(16,810) IDPV,NMRDPL(1),NMRDPL(2)
53      810 FORMAT("@B@S@X@077@Y@072",I4,"@X@077@Y@076",2A5)
54          ENDLIST
55    C     M1.2
56    C     NORMAL EXIT FROM PROGRAM
57    C
60      195 IF(.NOT.DEPRESS(16)) GO TO 200
61          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
10          IF(NFLAG(1).EQ.0) GO TO 500
11          MFLAG(2)=1
12          NO=1
13          NMDSPL(1)=5HENTER
14          NMDSPL(2)=4H IPV
15          IF(LFLG(3).NE.1) GO TO 100
16          IDPV=IINUM
17          IPV=IOCT(IINUM)
20          GO TO 40
21    C     M1.4
22    C     FILE NAME OPTION
23    C
24      500 IF(NFLAG(2).EQ.0) GO TO 510
25          MFLAG(1)=1
26          NO=1
27          NMDSPL(1)=5HENTER
30          NMDSPL(2)=5H FILE
31          JJJ=1
32          IF(LFLG(3).NE.1) GO TO 100
33          NAMER(1)=NME(1)
34          NAMER(2)=NME(2)
35          NMRDPL(1)=NME(1)
36          NMRDPL(2)=NME(2)
37          GO TO 40
40    C     M1.5
41    C     LIST VOLUME OPTION
42    C
43      510 IF(NFLAG(3).EQ.0) GO TO 520
44          NO=1
45          NMDSPL(1)=5HTBD
46          NMDSPL(2)=5HENTER
47          IF(LFLG(3).NE.1) GO TO 100
50          GO TO 40
51    C     M1.6
52    C     BLACK BOX OPTION
53    C
54      520 IF(NFLAG(4).EQ.0) GO TO 530
55          NFLAG(6)=0
56          IF((MFLAG(1).EQ.0).AND.(MFLAG(2).EQ.0)) GO TO 100
57          DO 525 I=1,2
60          NMB(I)=NAME(I,NO)
61      525 CONTINUE
62          DO 526 J=1,13
63          WT(J)=V(NO,J)
64          IF((J.LT.2).OR.(J.GT.4)) GO TO 526
65          I=J-1
66          PWT(I)=V(NO,J)
```

```
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)
```

```
1           DO 528 J=1,13
2           V(NO,J)=WT(J)
3           IF((J.LT.2).OR.(J.GT.4)) GO TO 528
4           I=J-1
5           DCG(NO,I)=DCG(NO,I)+(V(NO,J)-PWT(I))
6         528 CONTINUE
7           NFLAG(4)=0
10          NUP=1
11          NFLAG(6)=1
12          GO TO 50
13    C    M1.7
14    C    GEOMETRY OPTION
15    C
16        530 IF(NFLAG(5).EQ.0) GO TO 540
17          NFLAG(4)=NFLAG(6)=0
20          CALL WABCOM
21          NTOT=NTOT+1
22          NAME(1,NTOT)=NMBB(1)
23          NAME(2,NTOT)=NMBB(2)
24          DO 535 J=1,13
25          V(NTOT,J)=WT(J)
26        535 CONTINUE
27          DO 536 I=1,3
30          DCG(NTOT,I)=0.
31        536 CONTINUE
32          NFLAG(5)=0
33          NFLAG(6)=1
34          NUP=1
35          GO TO 50
36    C    M1.8
37    C    DISPLAY OPTION
40    C
41        540 IF(NFLAG(6).EQ.0) GO TO 550
42          IF(NUP.EQ.1) GO TO 543
43          IF(IPASS.NE.0) GO TO 544
44          NAMER(1)=NMRDPL(1)
45          NAMER(2)=NMRDPL(2)
46          CALL LOCFLE(IPV,NAMER,IFILNO)
47          OPEN(20,IPV,IFILNO)
50          READ(20,541) NTOT
51        541 FORMAT(I3)
52          READ(20,542) (NAME(1,I),NAME(2,I),VALUE1(I),VALUE2(I),VALUE3(I),
53         * VALUE4(I),VALUE5(I),VALUE6(I),VALUE7(I),VALUE8(I),VALUE9(I),
54         * VALUE10(I),VALUE11(I),VALUE12(I),VALUE13(I),IGNME(1,I),
55         * IGNME(2,I),(DCG(I,J),J=1,3),I=1,NTOT)
56        542 FORMAT(2A5,6F10.1,/,7F10.1,/,2A5,3F10.1)
57          CLOSE(20)
60        543 CALL WABSET
61        544 ITEMP=NTOT/LINES
62          IRMDR=NTOT//LINES
63          IF(IRMDR.NE.0) ITEMP=ITEMP+1
64          IF(.NOT.DEPRESS(32)) GO TO 546
65          IT=IT+1
```

```
1           IPASS=0
2         546 IF(IT.GE.ITEMP) IT=0
3           ISTRT=1+(IT*LINES)
4           IEND=LINES+(IT*LINES)
5           IF(IEND.GT.NTOT) IEND=NTOT
6           CALL WABLST
7           IF((MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)) GO TO 547
10          DO 545 I=1,12
11          IF(I.EQ.6) GO TO 545
12          IF(NFLAG(I).NE.1) GO TO 545
13          NFLAG(6)=0
14          GO TO 50
15        545 CONTINUE
16        547 NMDSPL(1)=NAME(1,NO)
17          NMDSPL(2)=NAME(2,NO)
20          IF(LFLG(3).EQ.0) GO TO 100
21          GO TO 50
22         M1.9
23    C    APPEND DISPLAY OPTION
24    C
25        550 IF(NFLAG(7).EQ.0) GO TO 560
26          IF(IX.EQ.6) GO TO 559
27          MFLAG(2)=1
30          IF(IX.NE.1) GO TO 552
31          MFLAG(1)=JJJ=1
32        552 NMDSPL(1)=5HENTER
33          NMDSPL(2)=IAPN(IX)
34          IF(LFLG(3).NE.1) GO TO 100
35          IF(IX.NE.1) GO TO 555
36          NTOT=NTOT+1
37          NAME(1,NTOT)=NME(1)
40          NAME(2,NTOT)=NME(2)
41          GO TO 557
42        555 J=IX-1
43          V(NTOT,J)=RRNUM
44          IF(J.EQ.1) GO TO 557
45          I=J-1
46          DCG(NTOT,I)=0.
47        557 IX=IX+1
50          GO TO 50
51        559 DO 558 J=5,13
52          V(NTOT,J)=0.
53        558 CONTINUE
54          DO 556 I=1,2
55          IGNME(I,NTOT)=5H
56        556 CONTINUE
57          IPASS=0
60          NUP=1
61          IX=1
62          NFLAG(7)=0
```

```
1            NFLAG(6)=1
2            GO TO 50
3       C    M1.10
4       C    DELETE COMPONENT OPTION
5       C
6         560 IF(NFLAG(8).EQ.0) GO TO 570
7            IF((MFLAG(1).EQ.0).AND.(MFLAG(2).EQ.0)) GO TO 100
10           NTOT=NTOT-1
11           DO 566 I=NO,NTOT
12           DO 564 J=1,2
13           NAME(J,I)=NAME(J,I+1)
14           IGNME(J,I)=IGNME(J,I+1)
15        564 CONTINUE
16           DO 565 K=1,13
17           V(I,K)=V(I+1,K)
20           IF((K.LT.2).OR.(K.GT.4)) GO TO 565
21           N=K-1
22           DCG(I,N)=DCG(I+1,N)
23        565 CONTINUE
24        566 CONTINUE
25           IPASS=0
26           NUP=1
27           NFLAG(8)=0
30           NFLAG(6)=1
31           GO TO 50
32      C    M1.11
33      C    STORE OPTION
34      C
35        570 IF(NFLAG(9).EQ.0) GO TO 580
36           NAMER(1)=NMRDPL(1)
37           NAMER(2)=NMRDPL(2)
40           CALL STORE(IPV,NAMER)
41           NFLAG(9)=0
42           NFLAG(6)=1
43           GO TO 50
44      C    M1.12
45      C    INPUT NEW DATA FILE OPTION
46      C
47        580 IF(NFLAG(10).EQ.0) GO TO 590
50           CALL INPUT(IPV)
51           GO TO 30
52      C    M1.13
53      C    BLACK BOX TRACKER OPTION
54      C
55        590 IF(NFLAG(11).EQ.0) GO TO 595
56           IF((MFLAG(1).EQ.0).AND.(MFLAG(2).EQ.0)) GO TO 100
57           NAMDT(1)=NAME(1,NO)
60           NAMDT(2)=NAME(2,NO)
61           XCG=V(NO,2)
62           YCG=V(NO,3)
63           ZCG=V(NO,4)
64           DO 593 J=2,4
65           I=J-1
66           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
```

```
1           DO 594 J=2,4
2           I=J-1
3           DCG(NO,I)=DCG(NO,I)+(V(NO,J)-PWT(I))
4         594 CONTINUE
5           NFLAG(11)=0
6           NUP=1
7           NFLAG(6)=1
10          GO TO 50
11    C     M1.14
12    C     PRINT OPTION
13    C
14        595 IF(NFLAG(12).EQ.0) GO TO 100
15          CALL $PPLCOPY
16          NFLAG(12)=0
17          NFLAG(6)=1
20          GO TO 50
21          SUBROUTINE RESET1
22    C     M2.0
23    C     FIRST INITIALIZATION OF MAIN PROGRAM VARIABLES
24    C
25          MFLAG(1)=MFLAG(2)=LFLG(10)=NO=NFLG=JJJ=0
26          LLL=KKK=MMM=NNN=IWRITE=IPRINT=0
27          INO=1
30          NUMO=9
31          DO 600 I=1,3
32        600 PWT(I)=0.
33          RETURN
34          SUBROUTINE RESET2
35    C     M3.0
36    C     SECOND INITIALIZATION OF MAIN PROGRAM VARIABLES
37    C
40          MFLAG(3)=IINUM=IPASS=IT=0
41          IX=1
42          RRNUM=0.
43          NMDSPL(1)=NMDSPL(2)=5H
44          DO 1 I=1,12
45          NFLAG(I)=0
46        1 CONTINUE
47          RETURN
50          END
```

```
1           SUBROUTINE MXM3X3(A,B,C)
2     C
3     C     C=AB    ALL 3X3 MATRICES
4     C
5           DIMENSION A(9),B(9),C(9)
6           C(1)=A(1)*B(1)+A(4)*B(2)+A(7)*B(3)
7           C(2)=A(2)*B(1)+A(5)*B(2)+A(8)*B(3)
10          C(3)=A(3)*B(1)+A(6)*B(2)+A(9)*B(3)
11          C(4)=A(1)*B(4)+A(4)*B(5)+A(7)*B(6)
12          C(5)=A(2)*B(4)+A(5)*B(5)+A(8)*B(6)
13          C(6)=A(3)*B(4)+A(6)*B(5)+A(9)*B(6)
14          C(7)=A(1)*B(7)+A(4)*B(8)+A(7)*B(9)
15          C(8)=A(2)*B(7)+A(5)*B(8)+A(8)*B(9)
16          C(9)=A(3)*B(7)+A(6)*B(8)+A(9)*B(9)
17          RETURN
20          END
```

```
1           SUBROUTINE MXV(M,V,VV)
2     C
3     C     THIS ROUTINE MULTIPLIES A 3X3 MATRIX TIMES A 3X1 MATRIX AND RETURNS
4     C     THE RESULTING 3X1 MATRIX.
5     C
6     C          INPUTS: M = 3X3 INPUT MATRIX
7     C                  V = 3X1 INPUT MATRIX (VECTOR)
10    C
11    C          OUTPUT: VV = 3X1 OUTPUT. MAY BE STORED IN SAME LOCATION
12    C                     AS INPUT VECTOR.
13    C
14          DIMENSION V(3),TMP(3),VV(3)
15          REAL M(9)
16          TMP(1) = M(1)*V(1) + M(4)*V(2) + M(7)*V(3)
17          TMP(2) = M(2)*V(1) + M(5)*V(2) + M(8)*V(3)
20          TMP(3) = M(3)*V(1) + M(6)*V(2) + M(9)*V(3)
21           DO 5 I=1,3
22           VV(I) = TMP(I)
23        5  CONTINUE
24          RETURN
25          END
```

```
1           SUBROUTINE PAT(A,W,R,B)
2       C
3       C  TRANSFORMS INERTIAS BY PARALLEL AXIS THEOREM
4       C
5           DIMENSION A(6),R(3),B(6)
6           B(1)=A(1)+W*(R(2)**2+R(3)**2)
7           B(2)=A(2)+W*(R(1)**2+R(3)**2)
10          B(3)=A(3)+W*(R(1)**2+R(2)**2)
11          B(4)=A(4)+W*R(2)*R(1)
12          B(5)=A(5)+W*R(1)*R(3)
13          B(6)=A(6)+W*R(3)*R(2)
14          RETURN
15          END
```

```
1     F      INCLUDE=FORMT
2            SUBROUTINE STORE(IPV,NAMER)
3     C    S1.0
4     C    STORE DATA TO DISK
5     C
6            COMMON/PARA/ LFLG(10),MFLAG(4),SUM,LETTER,NO,NFLG,INO,NFLAG(12)
7           * ,JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),IINUM,RRNUM
10           COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
11          * VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
12          * VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
13           COMMON/WB/ NMB(2),WT(15),TEMP(18),IWRITE,IJ,ITNM(2),TIXX(15),NUMO
14          * ,IPRINT,PWT(3),DCG(60,3),IGNME(3,60)
15           DIMENSION IBUF(3000)
16           REAL VALUE1,VALUE2,VALUE3,VALUE4,VALUE5,VALUE6,VALUE7,VALUE8,
17          * VALUE9,VALUE10,VALUE11,VALUE12,VALUE13
20           INTEGER NAME,NAMER(3)
21           DATA FMT2/1515151515B/,FMT3/7777777777B/,FMT4/7777700000B/
22           CALL BLDNAM(NAMER)
23           ENCODE(5,1000,IBUF(1)) NTOT,FMT2
24    1000   FORMAT(I3,A2)
25           J=2
26           DO 5000 I=1,NTOT
27           ENCODE(75,2000,IBUF(J)) NAME(1,I),NAME(2,I),VALUE1(I),VALUE2(I),
30          * VALUE3(I),VALUE4(I),VALUE5(I),VALUE6(I),FMT2
31    2000   FORMAT(2A5,6F10.1,A5)
32           ENCODE(75,3000,IBUF(J+15))VALUE7(I),VALUE8(I),VALUE9(I),
33          * VALUE10(I),VALUE11(I),VALUE12(I),VALUE13(I),FMT2
34    3000   FORMAT(7F10.1,A5)
35           ENCODE(45,7000,IBUF(J+30))IGNME(1,I),IGNME(2,I),(DCG(I,K),
36          * K=1,3),FMT2
37    7000   FORMAT(2A5,3F10.1,A5)
40           J=J+39
41    5000   CONTINUE
42           ENCODE(5,4000,IBUF(J))FMT4
43    4000   FORMAT(A5)
44           ENCODE(5,6000,IBUF(J+1))FMT3
45    6000   FORMAT(A5)
46           LENGTH=J
47           CALL STRTF(IPV,NAMER)
50           CALL APNDF(IBUF(1),LENGTH)
51           CALL APNDF(IBUF(J+1),1)
52           CALL CLOSF
53           RETURN
54           END
```

```
1           SUBROUTINE TOT(A,B,N)
2           DIMENSION A(1),B(1)
3           DO 1 I=1,N
4           A(I)=A(I)+B(I)
5           B(I)=0.
6         1 CONTINUE
7           RETURN
10          END
```

```
1     F      INCLUDE=DISK,TTY,PPL
2            SUBROUTINE WABCOM
3     C    W1.0
4     C    MASS PROPERTIES CALCULATIONS
5     C
6            COMMON/PARA/ LFLG(10),MFLAG(4),SUM,LETTER,NO,NFLG,INO,NFLAG(12),
7           * JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),IINUM,RRNUM
10           COMMON/WB/ NMB(2),WT(15),TEMP(18),IWRITE,IJ,ITNM(2),TIXX(15),NUMO
11          * ,IPRINT,PWT(3),DCG(60,3),IGNME(3,60)
12           COMMON/GRD/ BBOX(50),TEST(200),TEST3(50),OPDPL(100),BOX(20)
13           COMMON/LTR/ IFLAG(40)
14           COMMON/NMES/ NMBB(2)
15           COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
16          * VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
17          * VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
20           DIMENSION L1(3),XIN(4),YIN(4),ZIN(4),L2(3),D1(3),ZU(3)
21           DIMENSION YU(3),XU(3),TEMP2(9),DC(9),PMIP(9),PMI(9)
22           DIMENSION DELX(3),X0(3),XBAR(3),XBRF(3),GIN(100)
23           DIMENSION SIXX(15),D2R(9),XA(30),BIN(150)
24           DIMENSION YA(30),ZA(30),XB(30),YB(30),ZB(30)
25           DIMENSION FACT(10),NMDS(6),IFILE(3),IFILEN(3)
26           EQUIVALENCE (IYY,WT(6)),(IZZ,WT(7)),(IXY,WT(8)),(IXZ,WT(9))
27           EQUIVALENCE (IYZ,WT(10)),(S,WT(11)),( IXX,WT(5))
30           EQUIVALENCE (AREA,WT(13)),(VOL,WT(12)),(DELY,DELX(2))
31          * ,(DELZ,DELX(3))
32           EQUIVALENCE (FACL,FACT(1)),(FACI,FACT(2)),(FACW,FACT(3))
33           EQUIVALENCE (FACG,FACT(4)),(PSI,FACT(8)),(THETA,FACT(9))
34           EQUIVALENCE (PHI,FACT(10))
35           REAL L1,L2,IXX,IYY,IZZ,IXY,IXZ,IYZ
36           LOGICAL IFLAG,SUB,BLKBX,RFLAG,AFLAG,BFLAG
37           INTEGER STAT,ROOT,STNAME,STATT,OVL
40    C
41           DATA DELX/3*0./PSI,PHI,THETA/3*0./RHO,H/2*1./
42           DATA FACI/1./,FACW/1./,FACG/1./,FACL/12./,TEMP/18*0./
43           DATA IFLAG/40*.FALSE./,TIXX/15*0./,IPRINT/0/,IBASE/0/
44           DATA NMB(1)/4HCOMP/,STNAME/4HSUBT/,IWRITE/0/,IREFL/0/
45           DATA WT/15*0./,SUB/.FALSE./,IFLP/0/
46           DATA NME/'      ','      '/,NMB(2)/'      '/,OVL/110B/,IIPV/110B/
47           DATA NMDS/5HENTER,5H VALU,5HE    ,5HENTER,5H NAME,5H     /
50           DATA IFILE/5HDATA ,5H     ,5H     /
51    C
52    C   FORMAT STATEMENTS
53    C
54     1000 FORMAT(/,1X,2A5,/6H WT  = ,G13.7,/6H XCG =,G13.7,5X,
55          * 5HYCG =,G13.7,5X,5HZCG =,G13.7,/6H IXX =,G13.7,5X,
56          * 5HIYY =,G13.7,5X,5HIZZ =,G13.7,/6H IXY =,G13.7,5X,
57          * 5HIXZ =,G13.7,5X,5HIYZ =,G13.7)
60     1011 FORMAT(6H ASF =,G13.7,5X,5HVOL =,G13.7,5X,5HAFR =,G13.7)
61     1500 FORMAT(2(3F10.0,I1))
62     1700 FORMAT(A5)
63     1710 FORMAT(I5)
64    C
65    C
66           NUMO=9
```

```
67          IF(IJ.GT.1) GO TO 6000
70          NTIME=0
71     6000 IF(.NOT.SWITCH(1)) GO TO 6660
72          IF(IWRITE.EQ.1) CLOSE(20)
```

```
1     C   W1.1
2     C   INITIALIZE SUBROUTINE VARIABLES
3     C
4      6660 CALL RESET
5           IF(NFLAG(6).NE.1) GO TO 4
6           CALL RESET69
7           BLKBX=.TRUE.
10          GO TO 77
11        4 IF(NFLAG(4).NE.1) GO TO 5
12          SUB=.TRUE.
13          BLKBX=.TRUE.
14          IFLP=1
15          GO TO 78
16    C   W1.2
17    C   GEOMETRY INPUTS SETUP
20    C
21        5 BLKBX=.FALSE.
22          DISPLAY GEO
23          CALL RESET69
24        6 CALL RESET
25          LABEL(GIN)
26          ZSET(0.0F)
27          WRITE(16,108) NMB(1),NMB(2),IDPV,IFILE(1),IFILE(2),RHO
30      108 FORMAT("@B@X@050@Y@030",2A5,
31         * "@X@050@Y@034       ",I4,
32         * "@X@050@Y@040",2A5,
33         * "@X@050@Y@044",F10.3)
34          WRITE(16,109) H,FACL,FACI,FACW
35      109 FORMAT("@B@X@050@Y@050",F10.3,
36         * "@X@050@Y@054",F10.3,
37         * "@X@050@Y@060",F10.3,
40         * "@X@050@Y@064",F10.3)
41          WRITE(16,110) FACG,WTF,IREFL,IWRITE,SUB
42      110 FORMAT("@B@X@050@Y@070",F10.3,
43         * "@X@050@Y@074",F10.3,
44         * "@X@140@Y@030         ",I1,
45         * "@X@140@Y@034         ",I1,
46         * "@X@140@Y@040",L9)
47          WRITE(16,111) DELX(1),DELY,DELZ,PSI
50      111 FORMAT("@B@X@140@Y@044",F10.3,
51         * "@X@140@Y@050",F10.3,
52         * "@X@140@Y@054",F10.3,
53         * "@X@140@Y@060",F10.3)
54          WRITE(16,112) THETA,PHI,IBASE
55      112 FORMAT("@B@X@140@Y@064",F10.3,
56         * "@X@140@Y@070",F10.3,
57         * "@X@140@Y@074         ",I1)
60          ENDLIST
61       15 LABEL(TEST3)
62          ZSET(0.0F)
63          IF(.NOT.IFLAG(30)) GO TO 89
64          I=1
65          IF(IFLAG(31)) I=4
66          WRITE(16,92) NMDS(I),NMDS(I+1),NMDS(I+2)
```

```
 67      92 FORMAT("0B0X0060Y0056",3A5)
 70      89 IF(IFLAG(31)) GO TO 96
 71         IF(LFLG(1).EQ.1) GO TO 88
 72         WRITE(16,90) IINUM
 73      90 FORMAT("0B0X0070Y0064",I10)
 74         GO TO 93
 75      88 WRITE(16,91) RRNUM
 76      91 FORMAT("0B0X0070Y0064",F10.5)
 77         GO TO 93
100      96 WRITE(16,97) NME(1),NME(2)
101      97 FORMAT("0B0X0070Y0064",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,NMB(1)) NME(1)
110         ENCODE(5,1700,NMB(2)) NME(2)
111         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(1)) NME(1)
7           ENCODE(5,1700,IFILE(2)) NME(2)
10          IGNME(1,NTOT+1)=IFILE(1)
11          IGNME(2,NTOT+1)=IFILE(2)
12          DO 3200 I=1,3
13       3200 IFILEN(I)=IFILE(I)
14          CALL LOCFLE(IIPV,IFILEN,IFILNO)
15          GO TO 6
16        322 IF(.NOT.IFLAG(6)) GO TO 325
17          IF(LFLG(3).NE.1) GO TO 15
20          RHO=RRNUM
21          GO TO 6
22        325 IF(.NOT.IFLAG(7)) GO TO 326
23          IF(LFLG(3).NE.1) GO TO 15
24          H=RRNUM
25          GO TO 6
26        326 DO 327 I=1,4
27          J=I+7
30          IF(.NOT.IFLAG(J)) GO TO 327
31          IF(LFLG(3).NE.1) GO TO 15
32          FACT(I)=RRNUM
33          GO TO 6
34        327 CONTINUE
35          IF(.NOT.IFLAG(12)) GO TO 328
36          IF(LFLG(3).NE.1) GO TO 15
37          IREFL=IINUM
40          GO TO 6
41        328 IF(.NOT.IFLAG(13)) GO TO 329
42          IF(LFLG(3).NE.1) GO TO 15
43          IWRITE=IINUM
44          GO TO 6
45        329 IF(.NOT.IFLAG(14)) GO TO 330
46          IF(LFLG(3).NE.1) GO TO 15
47          SUB=.FALSE.
50          IF(NME(1).EQ.1HT) SUB=.TRUE.
51          GO TO 6
52        330 DO 331 I=1,3
53          J=I+14
54          IF(.NOT.IFLAG(J)) GO TO 331
55          IF(LFLG(3).NE.1) GO TO 15
56          DELX(I)=RRNUM
57          GO TO 6
60        331 CONTINUE
61          DO 332 I=8,10
62          J=I+10
63          IF(.NOT.IFLAG(J)) GO TO 332
64          IF(LFLG(3).NE.1) GO TO 15
65          FACT(I)=RRNUM
66          GO TO 6
```

```
 67      332 CONTINUE
 70          IF(.NOT.IFLAG(22)) GO TO 359
 71          IF(LFLG(3).NE.1) GO TO 15
 72          WTF=RRNUM
 73          GO TO 6
 74      359 IF(.NOT.IFLAG(23)) GO TO 358
 75          IF(LFLG(3).NE.1) GO TO 15
 76          IBASE=IINUM
 77          GO TO 6
100      358 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)
110          IFILE(2)=IGNME(2,NO)
111          DISPLAY BKBX
112      797 CALL RESET
113          LABEL(BIN)
114          ZSET(0,0F) -
115          WRITE(16,208) NMB(1),NMB(2),(WT(I),I=1,7)
116      208 FORMAT("@B@X@030@Y@030",2A5,
117         * "@X@0300@Y@034",F10.0,
120         * "@X@0300@Y@040",F10.0,
```

```
1          * "0X0110 0Y0040",F10.0,
2          * "0X0170 0Y0040",F10.0,
3          * "0X0030 0Y0044",F10.0,
4          * "0X0110 0Y0044",F10.0,
5          * "0X0170 0Y0044",F10.0)
6           WRITE(16,212) (WT(I),I=8,13)
7        212 FORMAT("0B 0X0030 0Y0050",F10.0,
10         * "0X0110 0Y0050",F10.0,
11         * "0X0170 0Y0050",F10.0,
12         * "0X0030 0Y0054",F10.0,
13         * "0X0110 0Y0054",F10.0,
14         * "0X0170 0Y0054",F10.0)
15          WRITE(16,209) FACL,DELX(1),FACI,DELY
16       209 FORMAT("0B 0X0050 0Y0064",F10.0,
17         * "0X0140 0Y0064",F10.0,
20         * "0X0050 0Y0070",F10.0,
21         * "0X0140 0Y0070",F10.0)
22          WRITE(16,210) FACW,DELZ,FACG,PSI
23       210 FORMAT("0B 0X0050 0Y0074",F10.0,
24         * "0X0140 0Y0074",F10.0,
25         * "0X0050 0Y0100",F10.0,
26         * "0X0140 0Y0100",F10.0)
27          WRITE(16,211) OVL,THETA,IWRITE,PHI,IPRINT,IBASE,IFILE(1),IFILE(2)
30       211 FORMAT("0B 0X0050 0Y0104      ",03,
31         * "0X0140 0Y0104",F10.0,
32         * "0X0050 0Y0110         ",I1,
33         * "0X0140 0Y0110",F10.0,
34         * "0X0050 0Y0114         ",I1,
35         * "0X0140 0Y0114         ",I1,
36         * "0X0050 0Y0120",2A5)
37          ENDLIST
40       798 LABEL(TEST3)
41          ZSET(0.0F)
42          IF(.NOT.IFLAG(30)) GO TO 213
43          I=1
44          IF(IFLAG(31)) I=4
45          WRITE(16,214) NMDS(I),NMDS(I+1),NMDS(I+2)
46       214 FORMAT("0B 0X0060 0Y0056",3A5)
47       213 IF(IFLAG(31)) GO TO 220
50          IF(LFLG(1).EQ.1) GO TO 215
51          WRITE(16,216) IINUM
52       216 FORMAT("0B 0X0070 0Y0064",I10)
53          GO TO 218
54       215 WRITE(16,217) RRNUM
55       217 FORMAT("0B 0X0070 0Y0064",F10.5)
56          GO TO 218
57       220 WRITE(16,221) NME(1),NME(2)
60       221 FORMAT("0B 0X0070 0Y0064",2A5)
61       218 ENDLIST
62       219 IF(.NOT.IFLAG(30)) GO TO 219
63          CALL AN7
64          IF(.NOT.IFLAG(29)) GO TO 799
65          IF(LFLG(3).NE.1) GO TO 798
66          ENCODE(5,1700,NMB(1)) NME(1)
```

```
 67          ENCODE(5,1700,NMB(2)) NME(2)
 70          GO TO 797
 71      799 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
10          SUB=.FALSE.
11          IF(NME(1).EQ.1HT) SUB=.TRUE.
12          GO TO 797
13      810 IF(.NOT.IFLAG(21)) GO TO 815
14          IF(LFLG(3).NE.1) GO TO 798
15          IWRITE=IINUM
16          IF(IWRITE.EQ.1) OPEN(20,OVL)
17          GO TO 797
20      815 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      820 CONTINUE
27          DO 825 I=8,10
30          J=I+17
31          IF(.NOT.IFLAG(J)) GO TO 825
32          IF(LFLG(3).NE.1) GO TO 798
33          FACT(I)=RRNUM
34          GO TO 797
35      825 CONTINUE
36          IF(.NOT.IFLAG(32)) GO TO 826
37          IF(LFLG(3).NE.1) GO TO 798
40          IPRINT=IINUM
41          NUMO=9
42          IF(IPRINT.EQ.1) NUMO=13
43          GO TO 797
44      826 IF(.NOT.IFLAG(33)) GO TO 828
45          IF(LFLG(3).NE.1) GO TO 798
46          IBASE=IINUM
47          GO TO 797
50      828 IF(.NOT.IFLAG(34)) GO TO 827
51          IF(LFLG(3).NE.1) GO TO 798
52          ENCODE(5,1700,IFILE(1)) NME(1)
53          ENCODE(5,1700,IFILE(2)) NME(2)
54          ENCODE(5,1700,IGNME(1,NO)) NME(1)
55          ENCODE(5,1700,IGNME(2,NO)) NME(2)
56          GO TO 797
57      827 IF(IFLAG(28)) GO TO 77
60          GO TO 798
61  C    W1.4
62  C    COMPONENT COUNTER
63  C
64       77 NTIME=NTIME+1
65       78 IF(NUMO.EQ.9) CALL $PTCH9
66  C W1.5
```

```
 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,IIPV,IFILNO)
10    C  READ BLANK CARD IN GEOMETRY FILE
11          READ(NUMI,1550) IDUMMY
12     1550 FORMAT(I1)
13    C
14          READ (NUMI,1500,END=400) X,Y,Z,STAT, XX,YY,ZZ,STATT
15           RFLAG = .FALSE.
16           GO TO 80
17       30  IF (RFLAG) GO TO 50
```

```
1            RFLAG = .TRUE.
2            X = XX
3            Y = YY
4            Z = ZZ
5            STAT = STATT
6            GO TO 60
7         50 RFLAG = .FALSE.
10           READ (NUMI,1500,END=400) X,Y,Z,STAT, XX,YY,ZZ,STATT
11        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
14            MC = M
15        80 M = 1
16            IF (STAT .EQ. 2) GO TO 150
17            IF (.NOT. BFLAG) GO TO 84
20        75 DO 81 J =1,MC
21             XA(J) = XB(J)
22             YA(J) = YB(J)
23        81   ZA(J) = ZB(J)
24        83 XB(1) = X
25           YB(1) = Y
26           ZB(1) = Z
27           GO TO 30
30        84 IF (AFLAG) GO TO 85
31            BFLAG = .TRUE.
32            GO TO 75
33        85 AFLAG = .FALSE.
34            GO TO 83
35       150 AFLAG = .TRUE.
36            BFLAG = .FALSE.
37            N = N+1
40       160 XA(M) = X
41           YA(M) = Y
42           ZA(M) = Z
43           GO TO 30
44       180 M = M + 1
45           IF (AFLAG) GO TO 160
46            XB(M) = X
47            YB(M) = Y
50            ZB(M) = Z
51           IF (STAT .NE. 3) GO TO 30
52        200 MMIN = MIN0 (M,MC) - 1
53            MC = M
54    C   W1 8
55    C   BEGIN COMPUTATION OF SURFACE ELEMENT CHARACTERISTICS
56    C
57           DO 2000 J= 1,MMIN
60           IPASS=0
61             XIN(1) =+DELX(1)+XA(J )
62             XIN(2) =+DELX(1)+XA(J +1)
63             XIN(3) =+DELX(1)+XB(J +1)
64             XIN(4) =+DELX(1)+XB(J )
65             YIN(1) =+DELY+YA(J )
66             YIN(2) =+DELY+YA(J +1)
```

```
1             YIN(3) =+DELY+YB(J +1)
2             YIN(4) =+DELY+YB(J )
3             ZIN(1) =+DELZ+ZA(J )
4             ZIN(2) =+DELZ+ZA(J +1)
5             ZIN(3) =+DELZ+ZB(J +1)
6             ZIN(4) =+DELZ+ZB(J )
7     C   w1.9
10    C   TRANSFORM DATA TO REFERENCE COORD SYSTEM
11    C
12          L1(1)=XIN(4)-XIN(1)
13          L1(2)=YIN(4)-YIN(1)
14          L1(3)=ZIN(4)-ZIN(1)
15    C
16          L2(1)=XIN(2)-XIN(1)
17          L2(2)=YIN(2)-YIN(1)
20          L2(3)=ZIN(2)-ZIN(1)
21    C
22          D1(1)=XIN(4)-XIN(2)
23          D1(2)=YIN(4)-YIN(2)
24          D1(3)=ZIN(4)-ZIN(2)
25    C
26          X0(1)=XIN(1)
27          X0(2)=YIN(1)
30          X0(3)=ZIN(1)
31          GO TO 5010
32    C
33     5020 CONTINUE
34          L1(1)=XIN(2)-XIN(3)
35          L1(2)=YIN(2)-YIN(3)
36          L1(3)=ZIN(2)-ZIN(3)
37    C
40          L2(1)=XIN(4)-XIN(3)
41          L2(2)=YIN(4)-YIN(3)
42          L2(3)=ZIN(4)-ZIN(3)
43    C
44          D1(1)=XIN(2)-XIN(4)
45          D1(2)=YIN(2)-YIN(4)
46          D1(3)=ZIN(2)-ZIN(4)
47    C
50          X0(1)=XIN(3)
51          X0(2)=YIN(3)
52          X0(3)=ZIN(3)
53          GO TO 5010
54    C
55     5010 CONTINUE
56          IPASS=IPASS+1
57    C
60    C
61    C
62          CALL MXV(D2R,L1,L1)
63          CALL MXV(D2R,L2,L2)
```

```
1            CALL MXV(D2R,D1,D1)
2            CALL MXV(D2R,X0,X0)
3     C
4            CALL CROSS(D1,L1,ZU,ZM)
5            CALL CROSS(ZU,L1,YU,YM)
6            XM=SQRT(L1(1)**2+L1(2)**2+L1(3)**2)
7            IF(XM.LE.0.) GO TO 2000
10           DO 5000 I=1,3
11           XU(I)=L1(I)/XM
12           YU(I)=YU(I)/YM
13           ZU(I)=ZU(I)/ZM
14           DC(3*I-2)=XU(I)
15           DC(3*I-1)=YU(I)
16           DC(3*I)=ZU(I)
17      5000 CONTINUE
20           CALL MXV(DC,L2,TEMP)
21           A=TEMP(2)
22           B=XM
23           C=TEMP(1)
24           S=A*B/2.
25           IF(S.LE.0.) GO TO 2000
26           WT(1)=S*RHO*H
27    C      W1.10
30    C      CENTROID OF TRIANGLE
31    C
32             XBAR(1)=(B+C)/3.
33             XBAR(2)=A/3.
34             XBAR(3)=0.
35    C      W1.11
36    C      INERTIA MATRIX IN PANEL COORDINATES
37    C
40           TMP=WT(1)/18.
41           PMIP(1)=TMP*A**2
42           PMIP(5)=TMP*(B**2-B*C+C**2)
43           PMIP(9)=PMIP(1)+PMIP(5)
44           PMIP(2)=-TMP*A*(B/2.-C)
45           PMIP(4)=PMIP(2)
46           PMIP(3)=0.
47           PMIP(6)=0.
50           PMIP(7)=0.
51           PMIP(8)=0.
52    C      W1.12
53    C      ROTATE INERTIA MATRIX TO PARALLEL TO REFERENCE FRAME
54    C
55             CALL MXM3X3(PMIP,DC,TEMP)
56             CALL XPOSE(DC,TEMP2)
57             CALL MXM3X3(TEMP2,TEMP,PMI)
60    C        W1.13
61    C        COMPUTE CENTROID POSITION IN REFERENCE FRAME
62    C
63             CALL MXV(TEMP2,XBAR,TEMP)
64             XBRF(1)=TEMP(1)+X0(1)
65             XBRF(2)=TEMP(2)+X0(2)
66             XBRF(3)=TEMP(3)+X0(3)
```

```
1     C   W1.14
2     C   VOLUME OF PANEL WRT X-Y PLANE AND AREA ON Y-Z PLANE
3     C
4           VOL=DC(9)*S*XBRF(3)
5           AREA=S*DC(3)
6           IF(AREA.LT.0.) AREA=0.
7     C   W1.15
10    C   MOMENT OF PANEL IN REF COORD
11    C
12          DO 5100 N=1,3
13          WT(N+1)=XBRF(N)*WT(1)
14     5100 CONTINUE
15    C   W1.16
16    C   COMPUTE MOMENTS AND PRODUCTS OF INERTIA IN REFERENCE FRAME
17    C
20          IXX=PMI(1)
21          IYY=PMI(5)
22          IZZ=PMI(9)
23          IXY=-PMI(2)
24          IXZ=-PMI(3)
25          IYZ=-PMI(6)
26          CALL PAT(IXX,WT(1),XBRF,WT(5))
27    C   W1.17
30    C   SUBTOTAL OF MASS PROPERTIES
31    C
32          CALL TOT(SIXX,WT,15)
33    C   W1.18
34    C   IS QUADRALATERAL COMPLETED ?
35    C
36          IF(IPASS.EQ.1) GO TO 5020
37     2000 CONTINUE
40          GO TO 80
41      400 CONTINUE
42          WRITE(10,6941)
43     6941 FORMAT(' MADE IT TO 400')
44    C   W1.19
45    C   SET DELTA CG'S FOR NEW BASE LINE
46    C
47          IF(IBASE.EQ.0) GO TO 4041
50          DO 4040 I=1,3
51     4040 DCG(NO,I)=0.
52          IBASE=0
53    C   W1.20
54    C   START SUMMATION OF BLACK BOXES
55    C
56     4041 IRF=1
57      403 CALL LOAD(SIXX,TEMP,15)
60      409 IF(TEMP(1).LT.1.E-4) GO TO 402
61          DO 405 I=2,4
62          J=14+I
63          IF(ABS(TEMP(I)).LT.1.) TEMP(I)=0.
64          TEMP(J)=TEMP(I)
65      405 TEMP(I)=TEMP(I)/TEMP(1)
66          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. IRF.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
10          IF((BLKBX).OR.(WTF.EQ.0.).OR.(IRF.EQ.3)) GO TO 469
11    C     W1.21
12    C     DENSITY CALCULATION FOR GEOMETRY OPTION
13    C
14          RHO=WTF/(TEMP(11)*H)
15          SIXX(1)=TEMP(1)=WTF
16          DO 470 I=5,10
17          SIXX(I)=TEMP(I)=TEMP(I)*RHO
20        470 CONTINUE
21        469 CONTINUE
22          DO 404 I=5,10
23        404 TEMP(I)=TEMP(I)/4633.056
24          FAC2=FACL*FACL
25          TEMP(11)=TEMP(11)/FAC2
26          TEMP(13)=TEMP(13)/FAC2
27          TEMP(12)=TEMP(12)/(FAC2*FACL)
30          IF(IWRITE.EQ.0) GO TO 4999
31    C     W1.22
32    C     WRITE RESULTS TO DISK
33    C
34          WRITE(20,1000) NMB(1),NMB(2),(TEMP(I),I=1,10)
35          WRITE(20,1011) (TEMP(I),I=11,13)
36       4999 IF((NFLAG(6).EQ.1).OR.(IFLP.EQ.1)) GO TO 410
37          IF((NFLAG(5).EQ.1).AND.(IRF.EQ.3)) GO TO 410
40    C     W1.23
41    C     WRITE RESULTS TO ANK OR PPL
42    C
43          WRITE(NUMO,1000) NMB(1),NMB(2),(TEMP(I),I=1,10)
44          WRITE(NUMO,1011) (TEMP(I),I=11,13)
45          IF(NUMO.EQ.13) CLOSE(NUMO)
46          IF(IRF.EQ.3) GO TO 407
47          NMBB(1)=NMB(1)
50          NMBB(2)=NMB(2)
51        407 IF((BLKBX).OR.(IRF.EQ.3)) GO TO 410
52          CALL LOAD(TEMP,WT,15)
53        410 IF(IRF.NE.3) GO TO 411
54          WRITE(10,6991) NFLAG(6)
55       6991 FORMAT(' NFLAG(6)=',I3)
56          IF(NFLAG(6).EQ.1) RETURN
57          IF(IFLP.EQ.1) GO TO 360
60          PAUSE
61          RETURN
62        411 IF(.NOT.SUB) GO TO 333
63          DO 334 JJ=1,13
64    C     W1.24
65    C     SUBTRACT COMPONENT'S MASS PROP. FROM TOTALS
66    C
```

```
 67      334 SIXX(JJ)=-SIXX(JJ)
 70      333 CONTINUE
 71          CALL TOT(TIXX,SIXX,15)
 72          IF(NFLAG(4).NE.1) GO TO 335
 73          NAME(1,NO)=NMB(1)
 74          NAME(2,NO)=NMB(2)
 75      335 IF(IFLP.EQ.1) GO TO 336
 76          ROOT=STNAME
 77          ENCODE(5,1700,NMB(1)) ROOT
100          ENCODE(5,1710,NMB(2)) NTIME
101      336 IRF=3
102          CALL LOAD(TIXX,TEMP,15)
103          GO TO 409
104  C    WI.25
105  C    BLACKBOX CONTRIBUTIONS
106  C
107     7000 CONTINUE
110          CALL LOAD(WT,SIXX,15)
111          DO 7010 I=1,10
112          FAC=FACI*4633.056
113          IF(I.LT.5) FAC=FACG
114          IF(I.EQ.1) FAC=FACW
115          SIXX(I)=SIXX(I)*FAC
116     7010 CONTINUE
117          SIXX(2)=SIXX(2)+DELX(1)
120          SIXX(3)=SIXX(3)+DELY
121          SIXX(4)=SIXX(4)+DELZ
122          CALL MXV(D2R,SIXX(2),SIXX(2))
123          PMIP(1)=SIXX(5)
124          PMIP(5)=SIXX(6)
125          PMIP(9)=SIXX(7)
126          PMIP(2)=-SIXX(8)
127          PMIP(4)=-SIXX(8)
130          PMIP(3)=-SIXX(9)
131          PMIP(7)=-SIXX(9)
132          PMIP(6)=-SIXX(10)
133          PMIP(8)=-SIXX(10)
134          CALL XPOSE(D2R,TEMP)
135          CALL MXM3X3(PMIP,TEMP,TEMP2)
136          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("0B0X0055 0Y0020GEOMETRY INPUTS")
24          CHAR(GIN)
25          PENON(1)
26          MENU("0B0X0020 0Y0030NAME       =0.0A0
```

```
1          * 8X80208Y8034INPUT PVV =8.8A8
2          * 8X80208Y8040GEOM FILE =8.8A8
3          * 8X80208Y8044RHO       =8.8A8
4          * 8X80208Y8050H         =8.8A8
5          * 8X80208Y8054FACL      =8.8A8
6          * 8X80208Y8060FACI      =8.8A8
7          * 8X80208Y8064FACW      =8.8A8
10         * 8X80208Y8070FACG      =8.8A8
11         * 8X80208Y8074FIX WT    =8.8A8
12         * 8X81108Y8030IREFL     =8.8A8
13         * 8X81108Y8034IWRITE    =8.8A8
14         * 8X81108Y8040SUB       =8.8A8
15         * 8X81108Y8044DELX      =8.8A8
16         * 8X81108Y8050DELY      =8.8A8
17         * 8X81108Y8054DELZ      =8.8A8
20         * 8X81108Y8060PSI       =8.8A8
21         * 8X81108Y8064THETA     =8.8A8
22         * 8X81108Y8070PHI       =8.8A8
23         * 8X81108Y8074BASE CHNGE=8.8A8
24         * 8X80708Y8104PROCEED8.8Z")
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          IFLAG(30)=.TRUE.
 73          GO TO 500
 74      311 IFLAG(14)=.TRUE.
 75          IFLAG(31)=.TRUE.
 76          IFLAG(30)=.TRUE.
 77          GO TO 500
100      312 IFLAG(15)=.TRUE.
101          IFLAG(30)=.TRUE.
102          GO TO 500
103      313 IFLAG(16)=.TRUE.
104          IFLAG(30)=.TRUE.
105          GO TO 500
106      314 IFLAG(17)=.TRUE.
107          IFLAG(30)=.TRUE.
110          GO TO 500
111      315 IFLAG(18)=.TRUE.
112          IFLAG(30)=.TRUE.
113          GO TO 500
114      316 IFLAG(19)=.TRUE.
115          IFLAG(30)=.TRUE.
116          GO TO 500
117      317 IFLAG(20)=.TRUE.
120          IFLAG(30)=.TRUE.
121          GO TO 500
122      298 IFLAG(22)=.TRUE.
123          IFLAG(30)=.TRUE.
124          GO TO 500
125      297 IFLAG(23)=.TRUE.
126          IFLAG(30)=.TRUE.
127          GO TO 500
130      299 IFLAG(21)=.TRUE.
131          IFLAG(30)=.TRUE.
132      500 CALLSIMG GRIDW
133          RETURN
134          IMAGE BKBX
135    C     W=.0
136    C     DISPLAY BLACK BOX INPUTS
137    C
140          CLEARPEN(1)
141          CHAR("@B@X@060@Y@020BLACK BOX INPUTS")
```

```
1           CHAR(BIN)
2           PENON(1)
3           MENU("&B&X&010&Y&030NAME= &.&A&
4          * &X&010&Y&034WT   = &.&A&
5          * &X&010&Y&040XCG = &.&A&
6          * &X&070&Y&040YCG = &.&A&
7          * &X&150&Y&040ZCG = &.&A&
10         * &X&010&Y&044IXX = &.&A&
11         * &X&070&Y&044IYY = &.&A&
12         * &X&150&Y&044IZZ = &.&A&
13         * &X&010&Y&050IXY = &.&A&
14         * &X&070&Y&050IYZ = &.&A&
15         * &X&150&Y&050IXZ = &.&A&
16         * &X&010&Y&054ASF = &.&A&
17         * &X&070&Y&054VOL = &.&A&
20         * &X&150&Y&054AFR = &.&A&
21         * &X&020&Y&064FACL       = &.&A&
22         * &X&020&Y&070FACI       = &.&A&
23         * &X&020&Y&074FACW       = &.&A&
24         * &X&020&Y&100FACG       = &.&A&
25         * &X&020&Y&104OVL        = &.&A&
26         * &X&020&Y&110IWRITE     = &.&A&
27         * &X&020&Y&114IPRINT     = &.&A&
30         * &X&020&Y&120GEOM FILE = &.&A&
31         * &X&110&Y&064DELX       = &.&A&
32         * &X&110&Y&070DELY       = &.&A&
33         * &X&110&Y&074DELZ       = &.&A&
34         * &X&110&Y&100PSI        = &.&A&
35         * &X&110&Y&104THETA      = &.&A&
36         * &X&110&Y&110PHI        = &.&A&
37         * &X&110&Y&114BASE CHNGE= &.&A&
40         * &X&070&Y&130PROCEED &.&Z")
41          PENOFF(1)
42          GO TO (364,365,366,367,368,369,370,371,372,373,374,375,376,377,
43         * 378,379,380,381,382,383,391,397,384,385,386,387,388,389,392,390)
44          GO TO 393
45       364 IFLAG(29)=.TRUE.
46           IFLAG(30)=.TRUE.
47           IFLAG(31)=.TRUE.
50           GO TO 393
51       365 IFLAG(3)=.TRUE.
52           IFLAG(30)=.TRUE.
53           GO TO 393
54       366 IFLAG(4)=.TRUE.
55           IFLAG(30)=.TRUE.
56           GO TO 393
57       367 IFLAG(5)=.TRUE.
60           IFLAG(30)=.TRUE.
61           GO TO 393
62       368 IFLAG(6)=.TRUE.
63           IFLAG(30)=.TRUE.
64           GO TO 393
65       369 IFLAG(7)=.TRUE.
66           IFLAG(30)=.TRUE.
```

```
 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
```

```
      376 IFLAG(14)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      377 IFLAG(15)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      378 IFLAG(16)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      379 IFLAG(17)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      380 IFLAG(18)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      381 IFLAG(19)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      382 IFLAG(20)=.TRUE.
          IFLAG(30)=.TRUE.
          IFLAG(31)=.TRUE.
          GO TO 393
      383 IFLAG(21)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      384 IFLAG(22)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      385 IFLAG(23)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      386 IFLAG(24)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      387 IFLAG(25)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      388 IFLAG(26)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      389 IFLAG(27)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      390 IFLAG(28)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      391 IFLAG(32)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      392 IFLAG(33)=.TRUE.
          IFLAG(30)=.TRUE.
          GO TO 393
      397 IFLAG(34)=.TRUE.
          IFLAG(31)=.TRUE.
```

```
 67          IFLAG(30)=.TRUE.
 70      393 CALLSIMG GRIDW
 71          RETURN
 72          SUBROUTINE RESET
 73    C     W4.0
 74    C     INITIALIZE SUBROUTINE VARIABLES AFTER EACH INPUT
 75    C
 76          IINUM=0
 77          RRNUM=0.
100          DO 395 I=1,40
101      395 IFLAG(I)=.FALSE.
102          RETURN
103          SUBROUTINE RESET69
104    C     W5.0
105    C     INITIALIZE SUBROUTINE VARIABLES AFTER EACH CALCULATION
106    C
107          IF(NFLAG(5).NE.1) GO TO 396
110          NMB(1)=5HDATA
111          NMB(2)=5HFILE
112          IDPV=110
113          IPV=110B
114          IFILE(1)=5H
115          IFILE(2)=5H
116      396 IREFL=IWRITE=IPRINT=0
117          WTF=DELX=DELY=DELZ=PHI=THETA=PSI=0.
120          RHO=H=FACI=FACW=FACG=1.
121          FACL=12.
122          SUB=.FALSE.
123          RETURN
124          END
```

```
      SUBROUTINE WABLST
C     L1.0
C     MENU COMPUTATION FOR COMPONENT DISPLAY
C
      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
      COMMON/CNT/ IPASS,NUP,NAMER(3),IPV,ISTRT,IEND,IT,LINES
      COMMON/OPT/ NHIT,OPDPL
      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 TOT1(20),TOT2(20),TOT3(20),TOT4(20),TOT5(20)
      INTEGER MNU1(200),MNU2(200),MNU3(200),MNU4(200),MNU5(200)
      INTEGER FMT(8),MASK(8),NAME,DX,DY,$PWD4,NME,MNU6(200)
      REAL VALUE1,VALUE2,VALUE3,VALUE4
      DATA NF/10/
      DATA MASK/8*1H /
C                1234567890123456789012345678901234567890123456789
      DATA FMT/'(9H("0S0B0X0,03,3H0Y0,03,     ,6H0.0A"))'/
      IF(NUP.EQ.1) GO TO 100
      IF(IPASS.NE.0) GO TO 900
  100 IPASS=IPASS+1
C     L1.1
C     COMPONENT'S NUMBER MENU
C
      DY=24
      DX=6
      LABEL(MNU6)
      ZSET(0.0F)
      DO 200 J=ISTRT,IEND
      FMT(8)=5H0."))
      IF(J.EQ.IEND) FMT(8)=5H0A"))
      FMT(6)=3H1I3
      ENCODE(40,FMT,MASK,H) DX,DY,J
      DY=DY+4
      WRITE(16,MASK)
  200 CONTINUE
C     L1.2
C     COMPONENT'S NAME MENU
C
      DY=24
      DX=10
      LABEL(MNU1)
      ZSET(0.0F)
      DO 300 J=ISTRT,IEND
      FMT(8)=5H0A"))
      IF(J.EQ.IEND) FMT(8)=5H0Z"))
      FMT(6)=3H2A5
      ENCODE(40,FMT,MASK,H) DX,DY,NAME(1,J),NAME(2,J)
      DY = DY+4
```

```
1           WRITE(16,MASK)
2         300 CONTINUE
3           ENDLIST
4       C   L1.3
5       C   TOTALS MENU
6       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=ISTRT,IEND
25          FMT(8)=5H0A"))
26          IF(J.EQ.IEND) FMT(8)=5H0Z"))
27          FMT(6)=5H
30          CALL DECFMT(VALUE1(J),NF,FMT(6))
31          ENCODE(40,FMT,MASK,H) DX,DY,VALUE1(J)
32          DY=DY+4
33          WRITE(16,MASK)
34        400 CONTINUE
35          ENDLIST
36      C   L1.5
37      C   TOTAL WEIGHT MENU
40      C
41          DY=DY+4
42          LABEL(TOT2)
43          ZSET(0.0F)
44          ENCODE(40,FMT,MASK,H) DX,DY,TEMP(1)
45          WRITE(16,MASK)
46          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=ISTRT,IEND
57          FMT(8)=5H0A"))
60          IF(J.EQ.IEND) FMT(8)=5H0Z"))
61          FMT(6)=5H
62          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)
66        500 CONTINUE
```

```
 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)=5H@A"))
112          IF(J.EQ.IEND) FMT(8)=5H@Z"))
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)
120      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)
```

```
1           DO 700 J=ISTRT,IEND
2           FMT(8)=5H0A"))
3           IF(J.EQ.IEND) FMT(8)=5H0Z"))
4           FMT(6)=5H
5           CALL DECFMT(VALUE4(J),NF,FMT(6))
6           ENCODE(40,FMT,MASK,H) DX,DY,VALUE4(J)
7           DY=DY+4
10          WRITE(16,MASK)
11     700  CONTINUE
12          ENDLIST
13    C   L1.11
14    C   TOTAL Z-CG MENU
15    C
16          DY=DY+4
17          LABEL(TOT5)
20          ZSET(0.0F)
21          ENCODE(40,FMT,MASK,H) DX,DY,TEMP(4)
22          WRITE(16,MASK)
23          ENDLIST
24          NUP=0
25     800  DISPLAY WABMNU
26     900  IF(LFLG(3).NE.1) RETURN
27          IF(JJJ.EQ.0) GO TO 1015
30          NAME(1,JJJ)=NME(1)
31          NAME(2,JJJ)=NME(2)
32          NUP=1
33          RETURN
34    1015  IF(KKK.EQ.0) GO TO 1030
35          VALUE1(KKK)=RRNUM
36          NUP=1
37          RETURN
40    1030  IF(LLL.EQ.0) GO TO 1045
41          PWT(1)=VALUE2(LLL)
42          VALUE2(LLL)=RRNUM
43          DCG(LLL,1)=DCG(LLL,1)+(VALUE2(LLL)-PWT(1))
44          NUP=1
45          RETURN
46    1045  IF(MMM.EQ.0) GO TO 1060
47          PWT(2)=VALUE3(MMM)
50          VALUE3(MMM)=RRNUM
51          DCG(MMM,2)=DCG(MMM,2)+(VALUE3(MMM)-PWT(2))
52          NUP=1
53          RETURN
54    1060  IF(NNN.EQ.0) GO TO 800
55          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 WABMNU
6     C   L2.0
7     C   DISPLAY COMPONENTS' MASS PROPERTIES
10    C
11          CHAR("8S8X8010 8Y8020  COMP                WT              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((LFLG(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          $PWD4=0
101          PENON(1)
102          MENU(MNU5)
103          PENOFF(1)
104          CHAR(TOT5)
105          IF((LFLG(10).EQ.1).OR.(MFLAG(1).EQ.1).OR.(MFLAG(2).EQ.1)
106         * .OR.(NHIT.EQ.1)) GO TO 1200
107          IF($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
```

```
1           SUBROUTINE WABSET
2     C     T1.0
3     C     ARRAY SETUP FOR WABCOM
4     C
5           COMMON/STR/ NTOT,VALUE1(60),VALUE2(60),VALUE3(60),VALUE4(60),
6          * VALUE5(60),VALUE6(60),VALUE7(60),VALUE8(60),VALUE9(60),
7          * VALUE10(60),VALUE11(60),VALUE12(60),VALUE13(60)
10          COMMON/PARA/ LFLG(10),MFLAG(4),SUM,LETTER,NO,NFLG,INO,NFLAG(12),
11         * JJJ,KKK,LLL,MMM,NNN,NAME(2,60),NME(2),IINUM,RRNUM
12          COMMON/WB/ NMB(2),WT(15),TEMP(18),IWRITE,IJ,ITNM(2),TIXX(15),NUMO
13         * ,IPRINT,PWT(3),DCG(60,3),IGNME(3,60)
14          DIMENSION V(60,13)
15          EQUIVALENCE (V(1,1),VALUE1(1))
16          DO 50 I=1,15
17          TIXX(I)=0.
20       50 CONTINUE
21          DO 100 IJ=1,NTOT
22          NMB(1)=NAME(1,IJ)
23          NMB(2)=NAME(2,IJ)
24          DO 200 IK=1,13
25          WT(IK)=V(IJ,IK)
26       200 CONTINUE
27          CALL WABCOM
30       100 CONTINUE
31          IF(IWRITE.EQ.1) CLOSE(20)
32          ITNM(1)=5HTOTAL
33          ITNM(2)=5HS
34          RETURN
35          END
```

```
1           SUBROUTINE XFORM(A,B,C,X)
2     C
3     C   THIS SUBROUTINE COMPUTES THE TRANSFORMATION MATRIX FOR A
4     C   YAW,PITCH,ROLL ROTATION SEQUENCE.
5     C
6     C           INPUTS:  A = YAW ANGLE IN DEGREES
7     C                    B = PITCH ANGLE IN DEGREES
10    C                    C = ROLL ANGLE IN DEGREES
11    C
12    C           OUTPUT:  X = 3X3 MATRIX OF TRANSFORMATION DIRECTION COSINES
13    C
14          DIMENSION X(9)
15          SA=SIN(A/57.295779)
16          CA=COS(A/57.295779)
17          SB=SIN(B/57.295779)
20          CB=COS(B/57.295779)
21          SC=SIN(C/57.295779)
22          CC=COS(C/57.295779)
23          X(1)=CB*CA
24          X(2)=SC*SB*CA-CC*SA
25          X(3)=CC*SB*CA+SC*SA
26          X(4)=CB*SA
27          X(5)=SC*SB*SA+CC*CA
30          X(6)=CC*SB*SA-SC*CA
31          X(7)=-SB
32          X(8)=SC*CB
33          X(9)=CC*CB
34          RETURN
35          END
```

```
1           SUBROUTINE XPOSE(A,B)
2     C
3     C      B = TRANSPOSE OF A     A & B ARE 3X3 MATRICES
4     C
5           DIMENSION A(9),B(9)
6           B(1)=A(1)
7           B(2)=A(4)
10          B(3)=A(7)
11          B(4)=A(2)
12          B(5)=A(5)
13          B(6)=A(8)
14          B(7)=A(3)
15          B(8)=A(6)
16          B(9)=A(9)
17          RETURN
20          END
```