AD NUMBER: AD0904981
LIMITATION CHANGES
TO: Approved for public release; distribution is unlimited.
FROM: Distribution limited to U.S. Government agencies only; Test and Evaluation; Jun 1972. Other requests for this document must be referred to the Air Force Armament Laboratory (DLRV). Eglin Air Force Base, FL 32542
AUTHORITY
AFAL ltr dtd 16 Dec 1975

AFATL-TR-72-129

TARGET DESCRIPTION AND VULNERABILITY PROGRAM

MARTIN MARIETTA CORPORATION

TECHNICAL REPORT AFATL-TR-72-129

JUNE 1972

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied June 1972. Other requests for this document must be referred to the Air Force Armament Laboratory (DLRV), Eglin Air Force Base, Florida 32542.

AIR FORCE ARMAMENT LABORATORY
AIR FORCE SYSTEMS COMMAND • UNITED STATES AIR FORCE
EGLIN AIR FORCE BASE, FLORIDA

Target Description
And
Vulnerability Program

George W. Brooks
Harvey N. Lerman

FOREWORD

This report summarizes the technical effort of Martin Marietta Corporation, Orlando, Florida, during the period 17 January through 19 May 1972 under Contract F08635-72-C-0103 with the Air Force Armament Laboratory, Eglin Air Force Base, Florida. The Armament Laboratory project engineer was TSgt Gary A. Brown (DLRV) Document No. OR 11,972 was assigned to the report by the contractor.

The Martin Marietta Corporation task leader for this program was Mr. George W. Brooks. Mr. Harvey N. Lerman was responsible for methodology formulation and the computer programming effort. Contributions were also made by Mr. George E. Belote III. This report was prepared by Messrs. George W. Brooks and Harvey N. Lerman under the direction of Mr. William R. Day, Chief of the Effectiveness Analysis Section, and Mr. Charles A. Borcher, Manager of the Warhead Technology Department.

Acknowledgement is made to TSgt Gary A. Brown and Mr. George Crews of the Air Force Armament Laboratory for their valuable technical assistance and guidance in the performance of this program.

This technical report has been reviewed and is approved.

CHARLES K. ARPKE, Lt Colonel, USAF
Chief, Weapons Effects Division

ABSTRACT

This program provided the Air Force an existing target description computer program which was demonstrated on the Eglin Air Force Base CDC 6600 computer by completing two check runs. The subsequent formulation, coding, and incorporation of two improvements in the program which provided a shielding capability and the input of standard geometrical shapes resulted in a modified program being adopted for use with the Eglin Air Force Base computer. The revised program was demonstrated by computing the presented area of the fuel tanks of a generic fighter aircraft, and making perspective plots of the target with hidden lines removed, using the SC4020 plotter.

This target description and vulnerability program also provides the Air Force with a unique tool for making rapid descriptive and vulnerability assessments of targets through the use of high speed computers and automatic plotting equipment.

(The reverse of this page is blank)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION I

INTRODUCTION AND SUMMARY

This final report summarizes a four-month technical effort to develop a computer program for performing rapid target description studies and assessments of the vulnerability of target components.

1. PROGRAM OBJECTIVES

The overall objective of this effort was to develop a computer program for rapid evaluation of targets and for use in performing target description and vulnerability studies. Specific tasks of the effort, as specified in Reference 1, were as follows:

1 Provide a copy of the existing contractor target description and vulnerability program.

2 Adapt the existing program to the Eglin Air Force Base CDC 6600 computer and SC4020 plotter.

3 Modify the program to provide:

- a A shielding capability in specifying the characteristics of target components.

- b Input of standard geometrical shapes.

4 Analyze two targets using the original program on the Eglin AFB computer.

5 Adapt the revised program to the Eglin AFB CDC 6600 computer and SC4020 plotter.

6 Analyze and describe one target to demonstrate the revised program on the Eglin computing equipment.

2. APPROACH SUMMARY

Previous methods used for target analysis were complicated and time-consuming, often requiring as much as 18 engineering man-months to complete and requiring the categorizing of up to one-half million critical bits of information. These methods precluded fast response to changing target spectra and indicated a need for a program to make quick target analyses.

In determining target vulnerability oata, a pictoral representation of the target is necessary. Such drawings can be used to show overall outlines of the targets as well as internal and external vulnerable components. As the target elements become complex and to easily obtain views from different angles, such drawings must be produced automatically rather than by laborious and time-consuming geometric projections. The combination of the computer and the plotter is a useful tool in generating the desired perspective drawings. In this new technique for describing a target, presented areas can be computed from these drawings for any viewing angle and the ratio of certain areas can be determined, such as the ratio of the presented area of vulnerable components to total presented area of the target. For example, in the case of simple targets with unprotected vulnerable component areas, this ratio provides an approximation for the probability of kill given a hit. In addition to the target plots obtained, with all hidden lines removed, the program computes presented areas for the perspective angles of interest.

3. ACHIEVEMENTS

During this effort, all technical requirements were achieved and resulted in a computer program which:

<u>1</u> Computes and plots presented areas of targets from any viewing angle.

<u>2</u> Accounts for shields and vulnerable components.

<u>3</u> Produces plots of the targets which have hidden lines removed.

<u>4</u> Accepts input in several forms depending on the target being evaluated.

A comprehensive user's guide was prepared, and several Air Force Armament Laboratory analysts were familiarized with program capabilities and utilization.

4. REPORT ORGANIZATION

This report discusses the methodology, capabilities, and utilization of this program. Section II describes the technical approach. Specific details of the program are contained in Section III. Section IV lists conclusions and recommendations for further work. The appendices contain flow charts, a program listing, common block descriptions, and input formats for using this program.

SECTION II

TECHNICAL APPROACH

1. METHODOLOGY

To develop a computerized technique for determining target and target component presented areas from different angles, simplified plan and side view drawings of the elements are necessary. In this approach complex shapes can be approximated with simple solid bodies, e.g., a parallelepiped may be used to represent an engine block of a truck. The coordinates (X, Y, Z) of the corners are determined for subsequent input into the computer program. Points to be connected by line segments can also be specified. The combination of solids, points, and line segments determines the various faces which approximate the target or component shapes. Since no provision is currently made for curved surfaces, they must be represented by a series of flat faces.

Having specified the target using the foregoing approximations, the presented area from any viewing angle can be computed. For a given target or group of components, only the outer defining lines are used to compute the presented area. Of more importance is the determination of average presented areas, i.e., presented areas at a given elevation angle but which have been averaged over all azimuth angles, since in an actual encounter the target orientation may not be reliably known. This is accomplished by taking the average area as seen from N views, where $N=2^{(K+1)}$, and K is incremented by 1 until the change in the average area is less than the specified tolerance. This definition of N arises from the fact that the views must be spaced symmetrically about the target, so that any increase in the number of views must be by a factor of two. The location of the viewer relative to the target must also be specified.

As an aid in computing presented areas, the perimeter of this target description program is plotted. Perspective plots of the target and vulnerable components are made with the hidden lines and extra lines used to define odd-shaped surfaces removed. These plots are made for only specified views and are not intended to reflect the averaged areas. In computing presented areas and producing perspective plots, the program applies transformation equations to the three-dimensional points and projects them into a plane.

Patios of areas may also be determined. For example, the ratio of the vulnerable area of a target to the total presented area would give an estimate of the single shot kill probability from that angle.

2. TARGET REPRESENTATION

The target to be run should be oriented so that if there is an axis of symmetry, it is the X axis. If there are two axes of symmetry, one should be X and the other Y.

Each part of the target should be redrawn so that it could be approximated by any one of the program's four standard solids (see Appendix I). Several solids may also be put together (but not touching) to form parts of the target. Any sections of the target which cannot be approximated accurately enough by use of standard solids must be described by means of face definitions. All points on the target which are either reference points of solids or part of a face description should be labeled for future reference.

The parts of the target should be grouped by object type, i.e., all parts which are shields should be denoted as type 2, while all vulnerable parts should be type 0. Parts which are neither shield nor vulnerable probably would be denoted as type 1, frame. An object's type can be changed from run to run, so the effect of adding or deleting a part (i.e., shield) can easily be determined.

SECTION III

COMPUTER PROGRAM

1. CAPABILITIES

This program can be used in many ways and can be the base for a more sophisticated system. Some of its capabilities are:

1 An object can be viewed from any point in space by providing elevation, azimuth, and range of the point from the object's origin,

2 An approach can be simulated by decreasing range on consecutive runs.

3 A movie can be generated by making small changes in elevation, azimuth, and/or range on consecutive runs.

4 An outer shell can be described and run as a frame in some runs and then run individually as a target. The ratio of the resultant areas would give the single-shot kill probability.

5 Other parametric studies could be made by changing object type (INOBJ) on card F of consecutive runs.

6 Open areas such as windows and doors can easily be described.

2. PROGRAM ORGANIZATION

The program consists of the following routines:

1 MAIN - Part 1 - Reads and generates the geometry,

Part 2 - Reads control card (F) and iterates on the azimuth until convergence.

2 ARCH 3 - For any given view this routine will generate the visible line segments of the target as well as the perimeter segments. The area is also calculated.

3 COVER - Determines whether a point is covered by a face in the viewing plane.

4 INTER - Calculates all edge intersections in the viewing plane.

5 SORTH - Sorting routine.

6 PLOTDD - Plotting routine.

7 ROTAT 3 - Three-dimensional point rotation routine.

The following perspective system routines are used:

1 INIT3D - Initializes the system for one perspective view.

2 XFORM - Projects the point onto the plane by calculating X and Y.

One other system routine is used:

DASHLN - Draws a dashed line between two points.

The following CRT routines from the Eglin library are used to replace the now dummy routine PLOTCC:

1 FRAMEV - Changes CRT frames.

2 LINEV - Draws a line on the CRT.

a. Perspective Equations

To produce perspective plots, transformation equations must be developed which, when applied to a point in three-space, will project it onto a plane. These equations can be easily derived and can be found in most texts on the subject. A summary of this derivation is explained below and shown in Figure 1.

The observer's eye is at point $\emptyset(\emptyset_X,\emptyset_Y,\emptyset_Z)$ and the line of sight, $\overline{\emptyset P}$, makes angles a, b, c, with the 3-space X-, Y-, Z- axis, respectively. $\overline{\emptyset D}$ is the perpendicular distance from the observer to the plane onto which point $P(P_X,P_Y,P_Z)$ is to be projected. If d is the distance $\overline{\emptyset D}$, the coordinates of D are

$$D_X = \emptyset_X + d\ (\cos a)$$

$$D_Y = \emptyset_Y + d\ (\cos b) \qquad (1)$$

$$D_Z = \emptyset_Z + d\ (\cos c).$$

The coordinates of the projected point P' in three-space are

$$P'_X = \emptyset_X + F\ (P_X - \emptyset_X)$$

$$P'_Y = \emptyset_Y + F\ (P_Y - \emptyset_Y) \qquad (2)$$

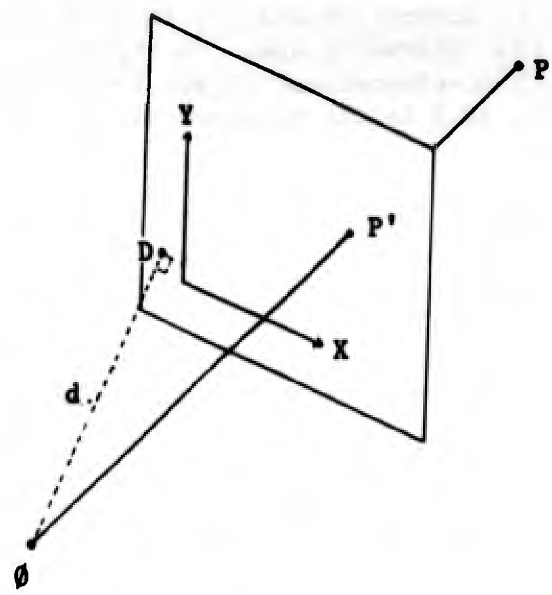


Figure 1. Geometry of Projection

$$P'_Z = Ø_Z + F\ (P_Z - Ø_Z)$$

where $F = d/\ (P_X - Ø_X)\cos a + (P_Y - Ø_Y)\cos b + (P_Z - Ø_Z)\cos c.$ (3)

The coordinates of P' on the new coordinate system then are

$$X = (P'_X - D_X)\cos b - (P'_Y - D_Y)\cos a\ /\sin c + X' \quad (4)$$

$$Y = (P'_Z - D_Z)/\sin c + Y'$$

where X', Y' is the distance from D to the origin of the new coordinate system.

Equations (1), (2), (3), and (4) can then be used to project any point in three-space (P) onto two-space (P').

The problem of scaling the object to be projected is readily solved by artificially moving the position of the observer's eye from point Ø to point Ø' such that $\overline{Ø'D} = d/f$ where f is the desired scale factor. Rotation can be accomplished similarly by rotating the observer's eye about the object.

Another way to accomplish scaling is by moving the projection plane away from or toward the observer's eye. In Figure 2, a rectangular solid is drawn around the three-dimensional object to be projected. The size of the solid is such that if A is the minimum corner of the solid and B is the maximum corner, then

$$A_X \leq P_X \leq B_X,$$

$$A_Y \leq P_Y \leq B_Y, \text{ and} \qquad (5)$$

$$A_Z \leq P_Z \leq B_Z$$

where P (P_X, P_Y, P_Z) is any arbitrary point within the object and A_X, A_Y, A_Z and B_X, B_Y, B_Z are the coordinates of A and B, respectively. If Ø is the observer's eye, rays ØA and ØB are the projection rays of the three-dimensional object. If the object is to be projected onto a plane so that it will fill an area ZDIM by YDIM, a rectangle of this size is moved, staying perpendicular to the line of sight from Ø so that two of the rectangles diagonally opposite corners A' and B' will fall on the two projection rays, ØA and ØB, respectively. The farther the rectangle is moved from Ø, the larger the projection will be. If the rectangle falls between the object and the observer, the projection will be scaled downward. But if the rectangle is moved to the far side of the object (A", B"), the projection will be scaled upward.

Windowing (viewing just part of the object) can be performed by using the rectangular solid originally chosen to enclose just the section of the object to be viewed. Then P(P_X, P_Y, P_Z) in Equation (5) becomes

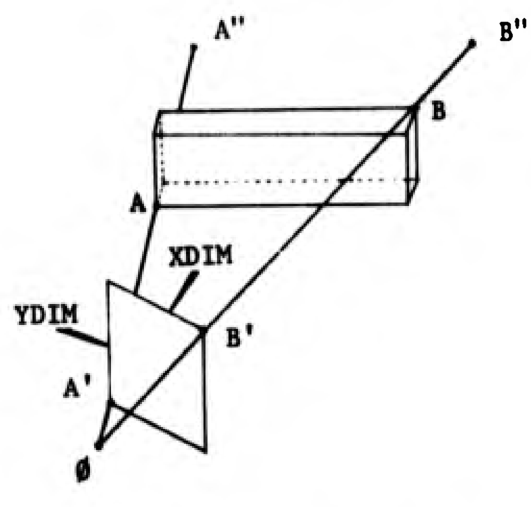


Figure 2. Geometry of Scaling

any arbitrary point within the window. Any other point, Q, within the object, but not the window, will be projected outside the projection rays and will not fall on the rectangle A'B'.

Once all the points in the object have been projected onto the plane, lines connecting these points can be drawn in the plane, thus projecting the whole object; however, the object is considered to be transparent, as all lines are drawn whether these lines can be seen by the observer or are actually hidden by some part of the object. The next important step is to determine which line segments are hidden from view. Then these segments can either be eliminated from the drawing or will show up as dashed.

b. Hidden Lines

One way of solving the hidden line problem requires that all surfaces in the three-dimensional object be approximated by flat planes which are called faces. The object can contain any number of faces, the boundaries of which are called edges, and must be approximated by straight lines. Edge ends are called vertices, and edges and vertices may be common to several faces or to only one face, as in describing an object shell. Faces may contain inner edges as well as outer edges, where the inner edges can define a window (a cutout or a transparent section). If several faces are coplanar, they can be combined into one face description. Using Equations (1) through (4), all the vertices in the object are projected onto the observation plane. By finding all edge intersections on this plane (Figure 3), the polyhedra problem can be simplified into the problem of determining whether a point is hidden by a face.

This simplification can readily be shown by observing that the edge connecting vertices V_1 and V_2 is composed of segments V_1I_1, I_1I_2, I_2I_3, and I_3V_2, where the points I_i are all the intersection points between this edge and all other edges. Since the I_i's are all the intersections, there can be no intersections on the individual segments. The visibility of segment j is equivalent to the visibility of any point in segment j, since visibility can change only at a point of intersection. Hence, if P_j is visible, segment j is visible.

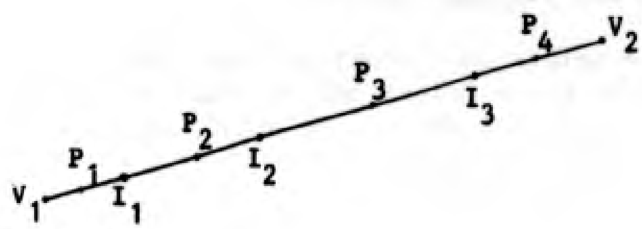
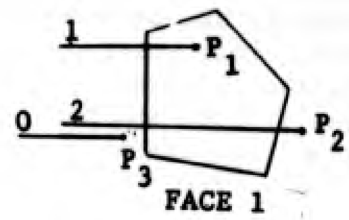


Figure 3. Edge Intersections

In order to determine the visibility of a point, one must first define the property of being covered. A point is said to be covered by a face if, in the observer's plane, the point falls within the area enclosed by the edges of the face. This property is easily determined by drawing a horizontal line through and to the left of the point and counting the number of edges crossed. (If an edge vertex is crossed, it is counted only if the edge is below the point.) If this number is odd, the point is covered; otherwise, it is not. Figure 4 contains some examples of determining the coverage of a point. Points P_1 and P_5 are shown to be covered by faces 1 and 2, respectively, while the other points are not covered.

Since faces are closed polygons, the total number of horizontal crossovers (to right as well as left) must be an even number (or zero). If an even number of crossovers is on the left, there also must be an even number on the right. And if an odd number of crossovers is on the left, there must be an odd number on the right. This property is observable as long as a face is described as a series of connected edges where the last vertex in the last edge is the same as the first vertex in the first edge (i.e., a closed figure). If face windows are described with this in mind, coverage is still observable. In Figure 5 the face is described by this ordered set of points 1, 2, 3, 4, 5, 1, 6, 7, 8, 9, 6, 1. Since face windows must also be closed polygons, the last point must be the same as the first. This causes the line between points 1 and 6 to be defined twice; as line 1, 6 and line 6, 1. This double line is called a pseudoedge and is ignored during actual plotting. These double lines can be thought of as a narrow cut allowing the window to be part of the outside. In this example, points P_2, P_4, and P_5 are covered while points P_1 and P_3 are not. The algorithm allows the window point, P_3 to show through.



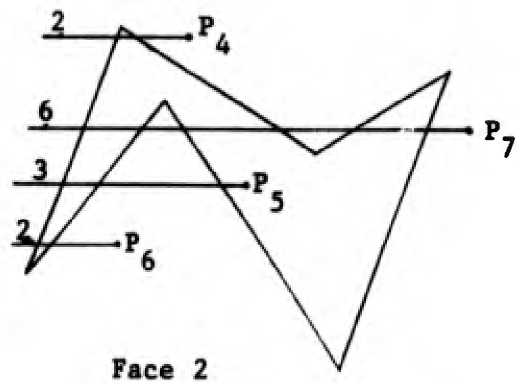


Figure 4. Coverage Geometry

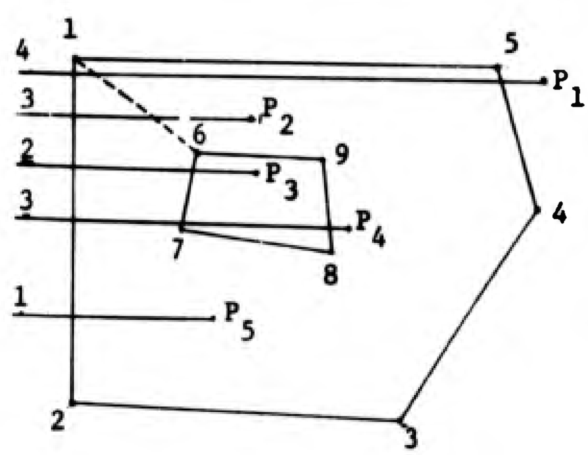


Figure 5. Window Coverage Geometry

If a point is not covered by any face, it is visible with no further tests required. But once it has been determined that a point is covered by a face, in order for that point to be visible it must be on the same side of the face as the observer's eye. This can be determined by putting the coordinates of both the point P_j and the observer's eye $\emptyset$ in the equation of the plane containing the face. The results, D_{P_j} and $D_\emptyset$, are the signed distance from the face to P_j and $\emptyset$, respectively. If D_{P_j} and $D_\emptyset$ have the same sign (i.e., $D_{P_j} * D_\emptyset > 0$), the point is visible. Otherwise, it is not.

The normal to the plane is determined by the cross-product of two of the edges in the plane. If P_1 (X_1, Y_1, Z_1), P_2 (X_2, Y_2, Z_2), and P_3 (X_3, Y_3, Z_3) are three vertices of the face and V_1 and V_2 are the vectors connecting these points, respectively, then

$$V_1 \times V_2 = A_i + B_j + C_k \tag{6}$$

where $A = a_{12}\, a_{23} - a_{13}\, a_{22}$

$B = a_{13}\, a_{21} - a_{11}\, a_{23}$

$C = a_{11}\, a_{22} - a_{12}\, a_{21}$

and

$$a = \begin{bmatrix} (X_1 - X_2) & (Y_1 - Y_2) & (Z_1 - Z_2) \\ (X_1 - X_3) & (Y_1 - Y_3) & (Z_1 - Z_3) \end{bmatrix}.$$

The equation of the plane containing this face is

$$A_x + B_y + C_z + D = 0 \tag{7}$$

where D is the distance from the origin to the plane. So

$$D = -(A_{x1} + B_{y1} + C_{z1}). \tag{8}$$

The distance from the observer's eye to the plane is $D_\emptyset = A_{\emptyset x} + B_{\emptyset y} + C_{\emptyset z} + D$ and the distance from point P_j to the plane is

$$D_{P_j} = A_{P_{j_x}} + B_{P_{j_y}} + C_{P_{j_z}} + D.$$

If $D_\emptyset * D_{P_j} > 0$, the point is visible.

If $D_\emptyset * D_{P_j} < 0$, the point is hidden

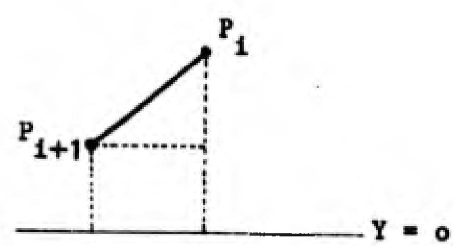
and the visibility of P_j with respect to $\emptyset$ is determined.

Since the visibility of P_j determines the visibility of segment j, the three-dimensional object can be drawn in perspective with the hidden lines identified and either eliminated or dashed.

c. Projection Area

Once an object has been drawn in perspective, its projected area can be found by making use of the same covered property described before. It has been shown that to be visible a line segment either must not be covered by any face, or if covered, must be in front of that face. A line segment which meets the first of these conditions is denoted as a perimeter segment. The collection of perimeter segments generated in this manner can be arranged to form a closed polygon called the projection or "shadow" of the object.

The area covered by this polygon can be found by an integration technique shown in Figure 6. The area under the line from P_i to P_{i+1} is the sum of the areas of the square and the triangle.



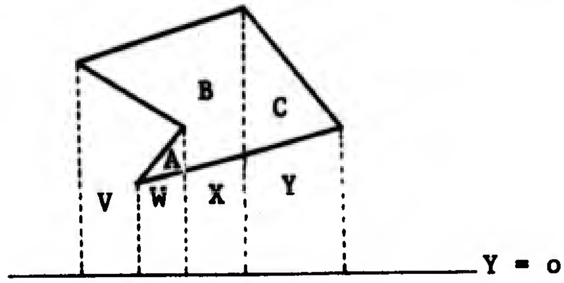
$$\text{AREA} = \left(X_i - X_{i+1}\right) Y_{i+1} + \frac{1}{2}\left(X_i - X_{i+1}\right)\left(Y_i - Y_{i+1}\right)$$
$$= \left(X_i - X_{i+1}\right) \frac{\left(Y_i + Y_{i+1}\right)}{2}$$

Figure 6. Integration Methodology

To find the area of the projected polygon, the equation in Figure 6 can be applied to every line segment in the perimeter, giving

$$\text{Total Area} = 1/2 \left| \sum_{i+1}^{n} (X_i - X_{i+1})(Y_i + Y_{i+1}) \right|. \tag{9}$$

Since the polygon is closed, the areas not enclosed will cancel out (see Figure 7).

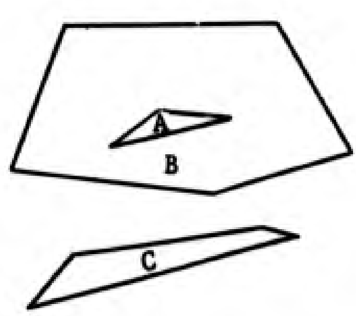


$$\text{Enclosed Area} = (C+Y) + (A+B+V+W+X) - (A+V+W) + (A+W) - (W+X+Y)$$

$$= A+B+C$$

Figure 7. Area Computation Methodology

Some objects may produce disjointed and/or window perimeters. Equation (9) is applied to each perimeter separately and then combined by a third application of the cover property. In Figure 8 the areas enclosed by the three perimeters are (A+B), A, and C.



$$\text{Area} = (A + B) + C - A = B + C$$

Figure 8. Disjointed and Window Perimeters

If a perimeter is covered by any other perimeter polygon, it encloses negative area. If not covered, it encloses positive area. In the example shown in Figure 8, the perimeter enclosing A is covered by the polygon (A + B). The other two perimeters are not covered. The resultant area sum is then

$$(A + B) + C - A = B + C.$$

Figures 9 through 15 show the overall flow diagrams of the program. Detailed flow diagrams are contained in Appendix II.

3. INPUT FORMAT

The target must be drawn on a grid in two orthogonal views to determine the coordinate of the end points of the line segments that define a face of the target. The input needed is shown below (also see Appendix I).

Card A

Columns	Contents	
1 - 5	J-Number of points in figure ≤ 600, J ≠ 0	
8	ID3-Debug flag e	
10	IØ - Offset flag	
11 - 15	Offset X	Used on all points, if IØ ≠ 0
16 - 20	Offset Y	
21 - 25	Offset Z	
26 - 30	FC-Factor converting input units to engineering units	
35	JDID - Flag (not used on CRT plotter)	

Card B

Columns	Contents	
1 - 5	X1	As many cards as required to describe the J points; the order of the points determines the point number used on card D. Range is determined from these points unless JDID ≠ 0. All points given need not be used on a D card. If IØ ≠ 0, each ordinate is offset by the appropriate value.
6 - 10	Y1	
11 - 15	Z1	
16 - 30	X2, Y2, Z2	
31 - 45	X3, Y3, Z3	
46 - 60	X4, Y4, Z4	
61 - 75	X5, Y5, Z5	

The objects to be presented may be described in several ways:

1 Describe each face in the object in terms of the points on card B,

2 Describe a solid object in terms of its size, location, and orientation.

3 Repeat, relocate, and reorient a previously described object.

MAIN

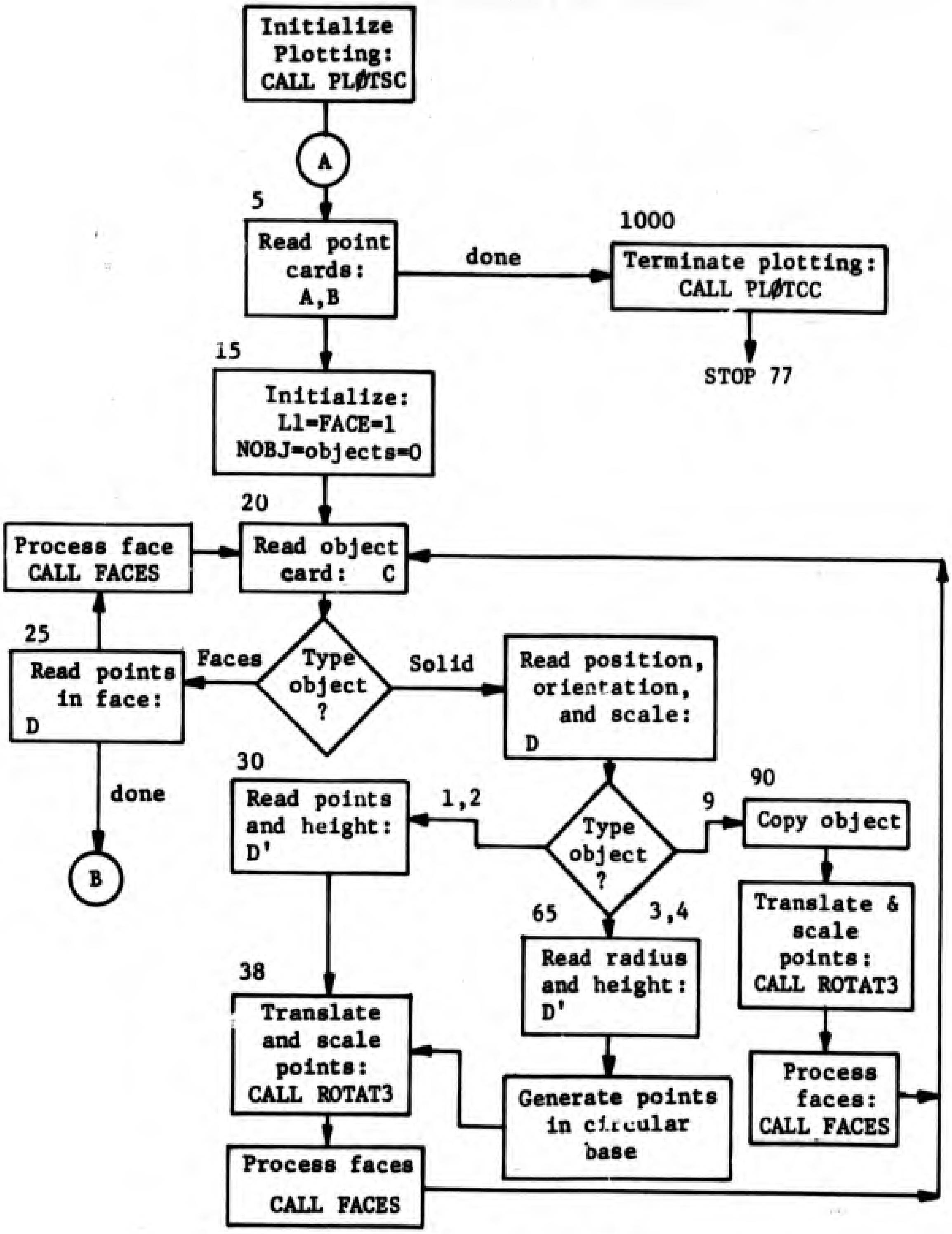


Figure 9. Subroutine MAIN

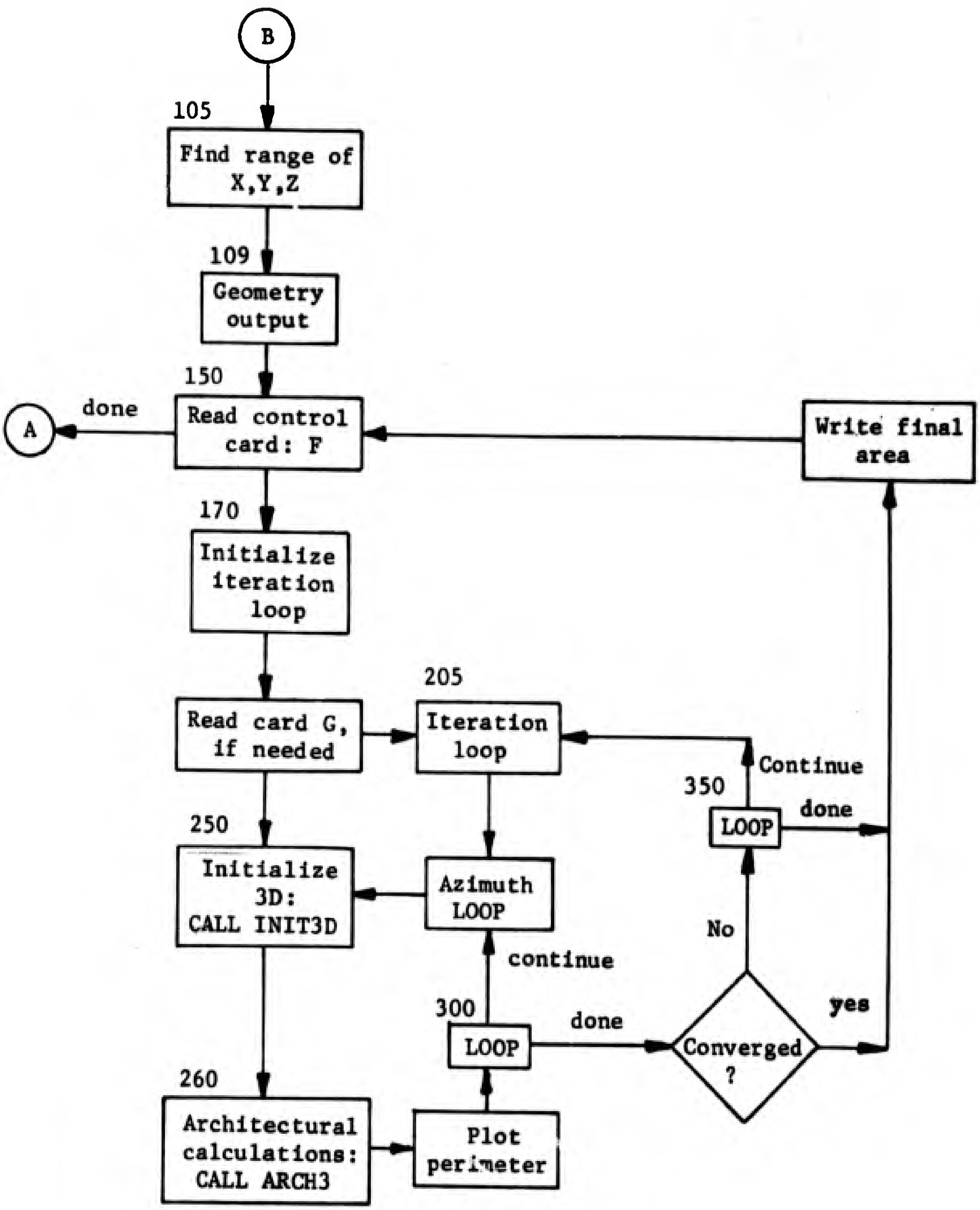


Figure 9. (Concluded)

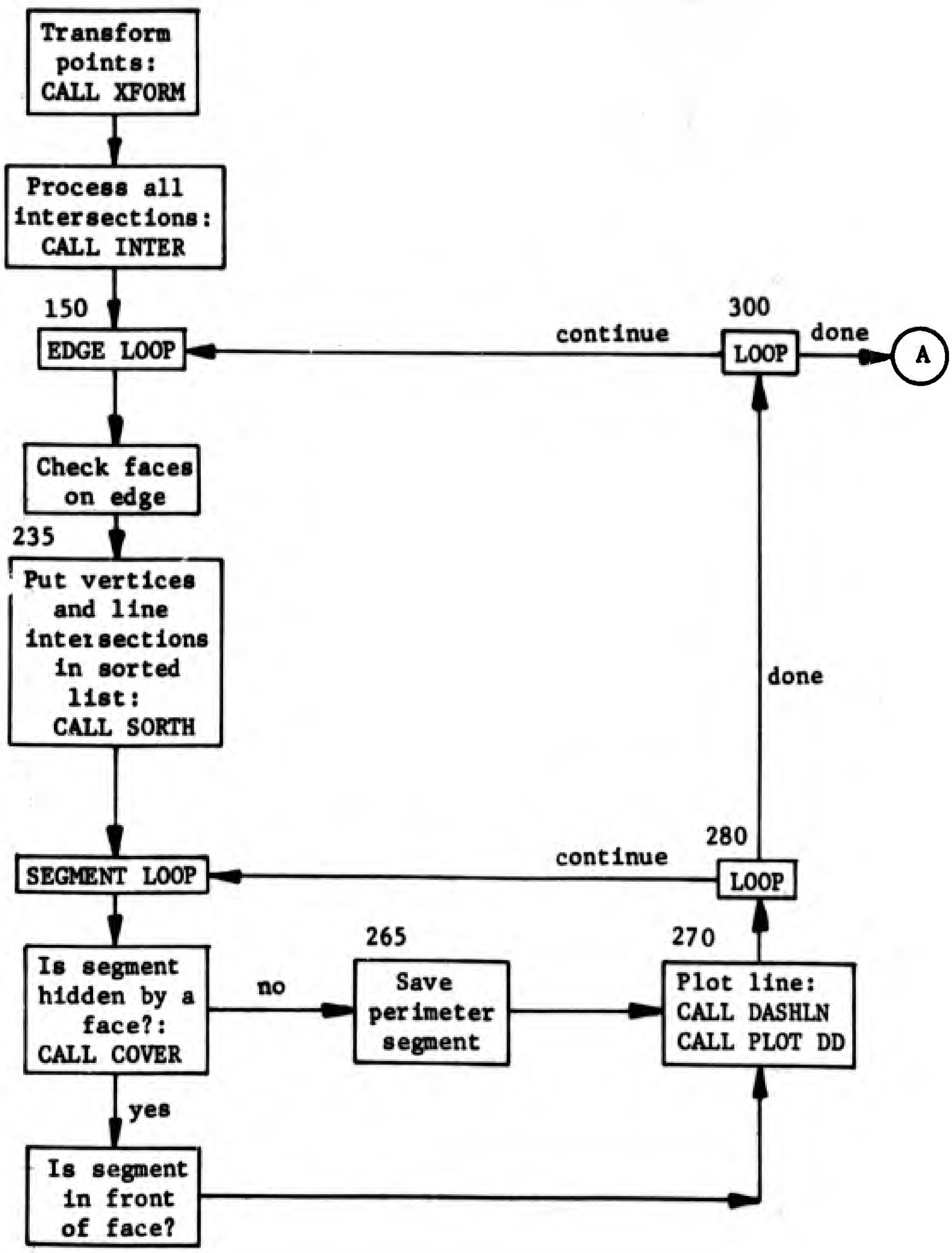


Figure 10. Subroutine ARCH3

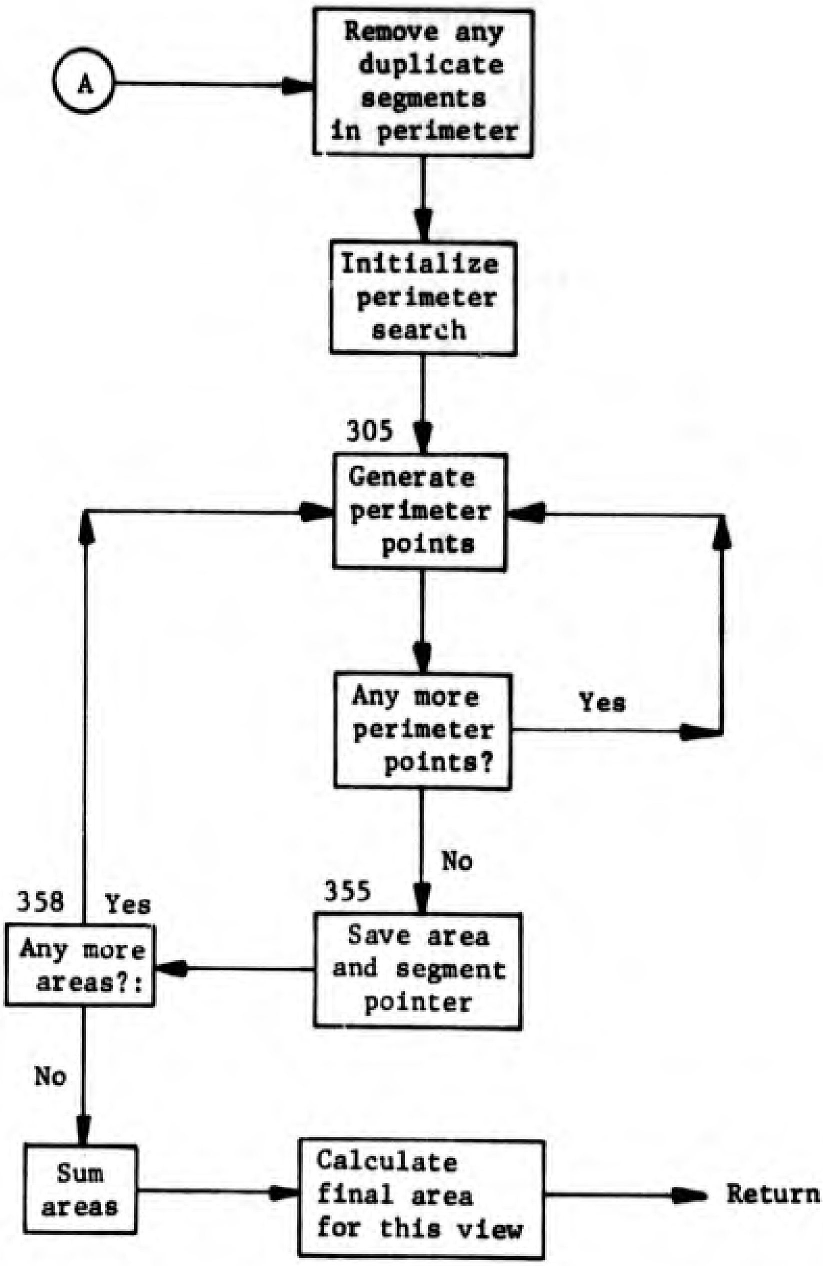


Figure 10. (Concluded)

COVER

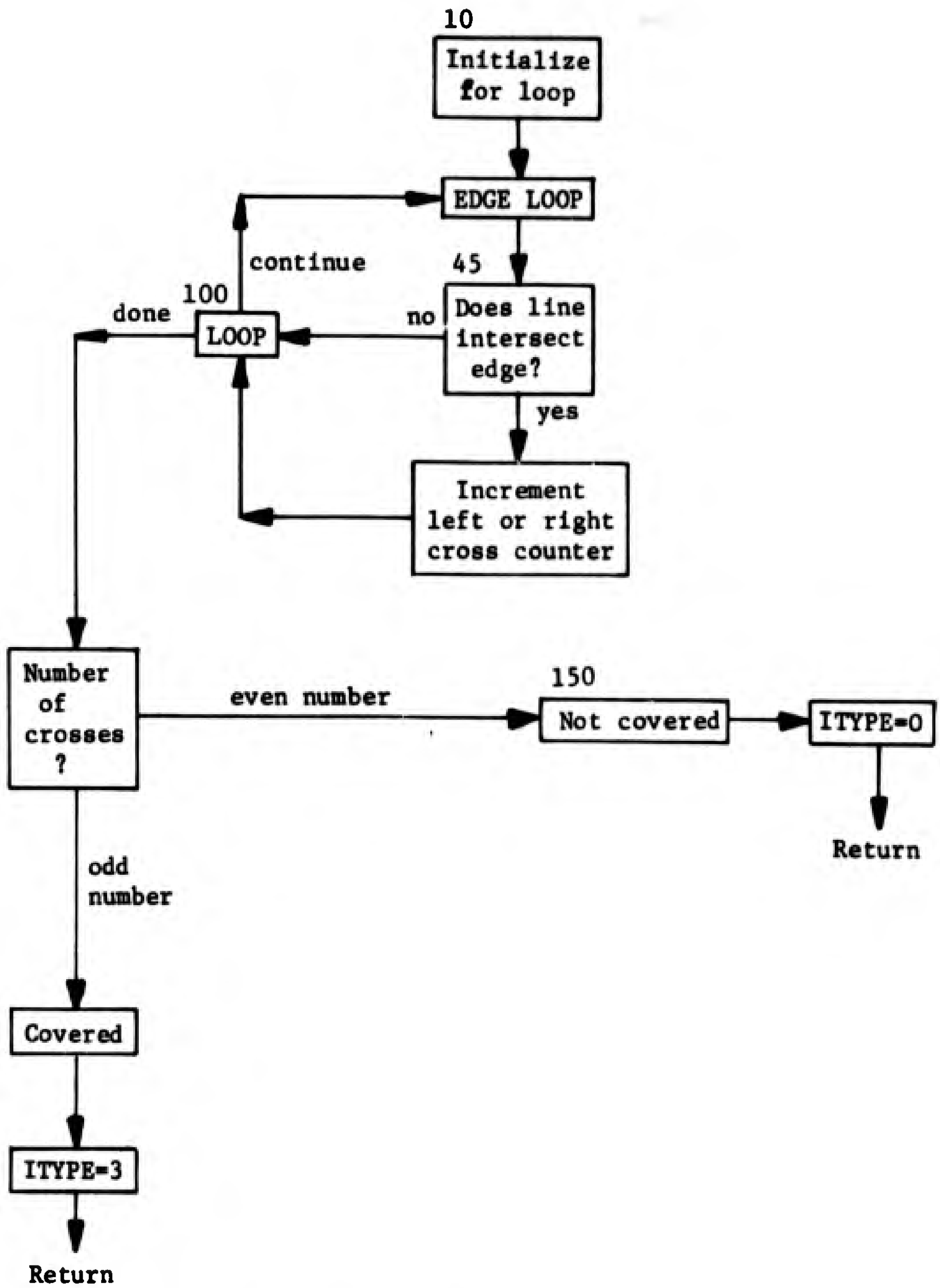


Figure 11. Subroutine COVER

INTER

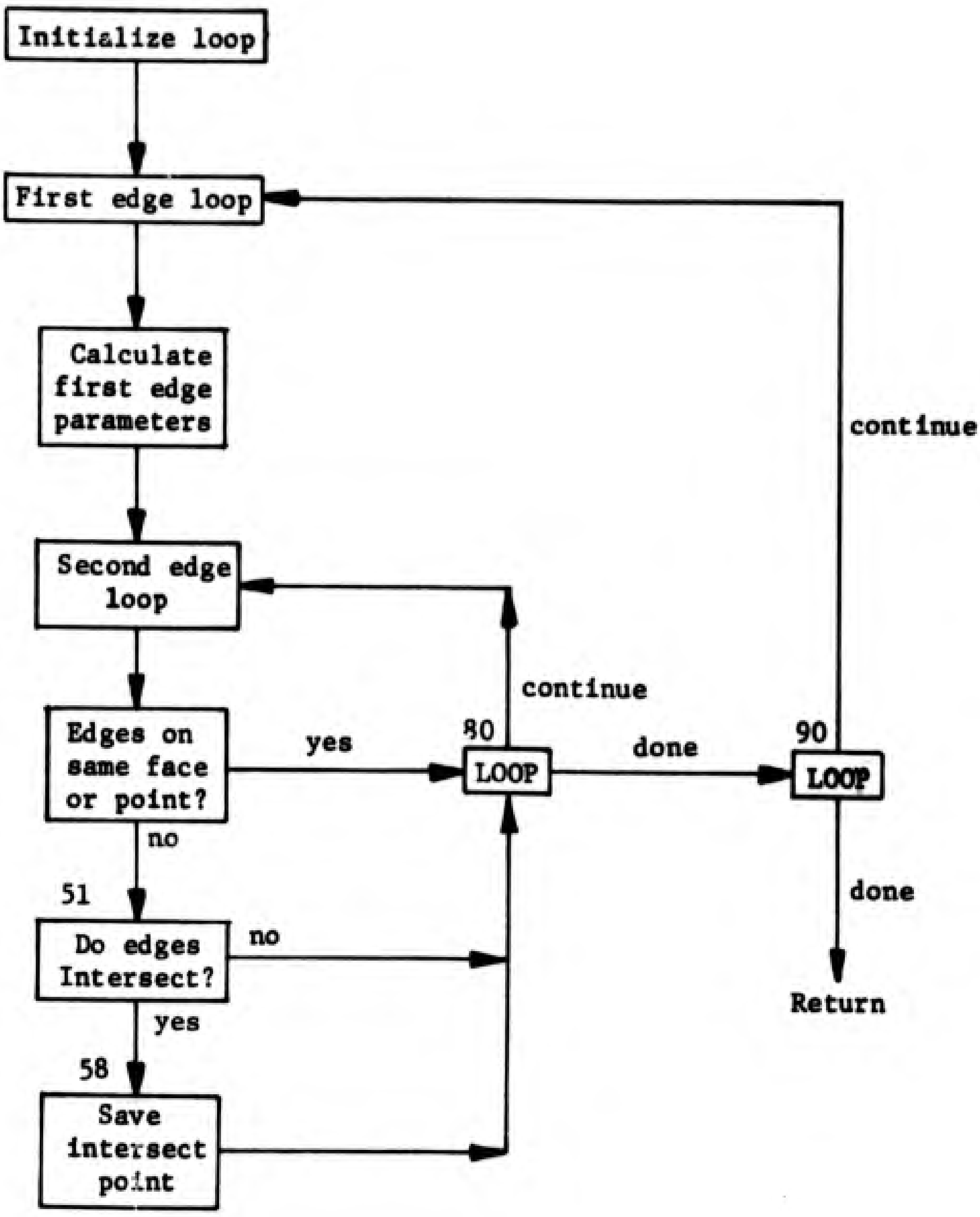


Figure 12. Subroutine INTER

SORTH

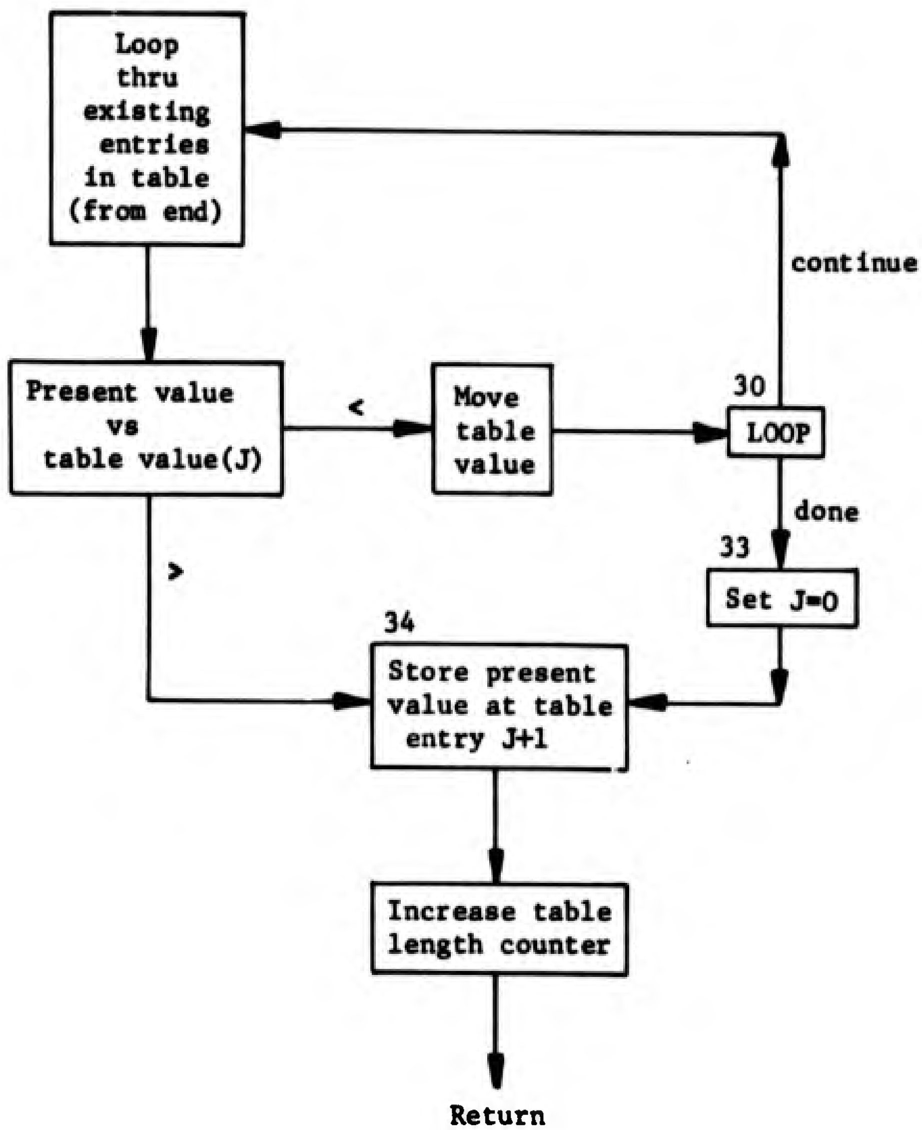


Figure 13. Subroutine SORTH

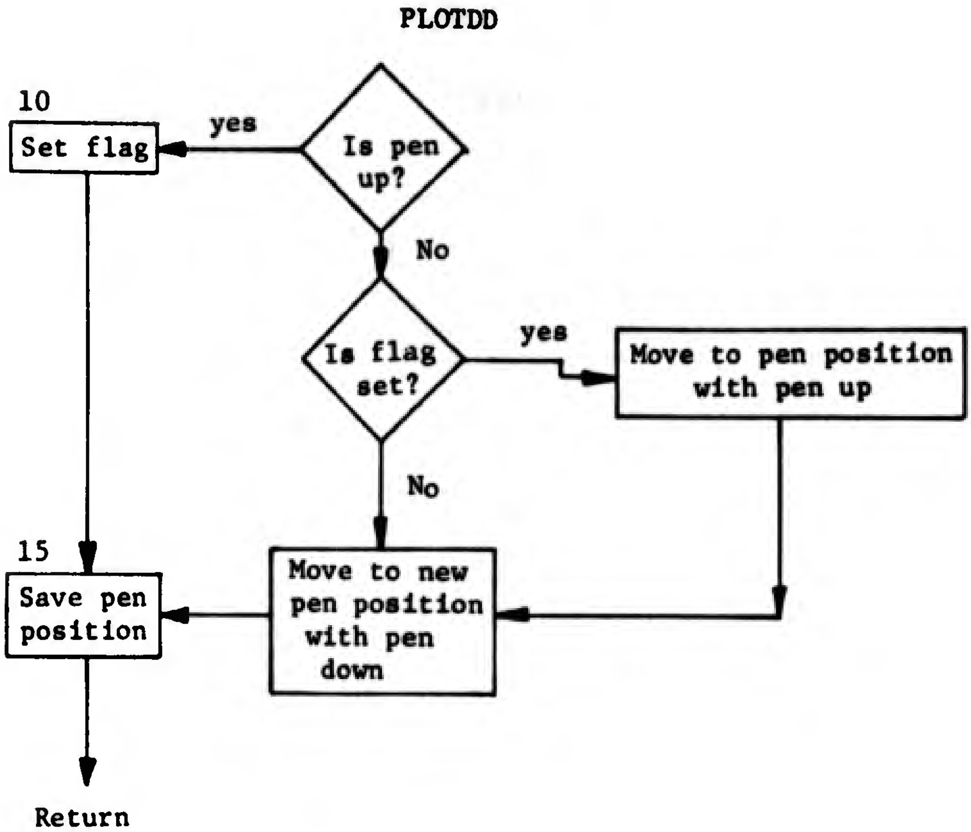


Figure 14. Subroutine PLOTDD

ROTAT3

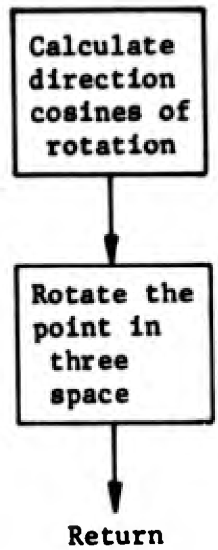


Figure 15. Subroutine ROTAT3

a. Faces

Card C

```
Columns      1 :  0 (or blank)
         2 - 5 :  N - Number of points in the following face ≤ 25
         9 - 10:  JØBJ - Non-blank, if this face is the first face in
                  a new object.
```

Card D

```
Columns 1 - 3 :  N1  ⎫
        4 - 6 :  N2  ⎪ Contiguous point numbers describing the face.
          .      .   ⎬ There will be N numbers on this card.  If an
          .      .   ⎪ edge is included more than once, it is a
          .      .   ⎪ pseudo-edge and is not drawn.  Ni is the ith
      73 - 75:  N25  ⎭ point on card B.
```

There must be a card C and D for each face to be described ≤ 300.

b. Solid

Card C

```
Columns      1:  K - Type solid
         2 - 5:  N - Number of points in the following object
                     base ≤ 20
        9 - 10:  JØBJ - Object number (must be one number greater
                 than previous number) (If blank, this solid is
                 part of previous object)
```

Card D

```
Columns 1 - 5 :  Xo ⎫
        6 - 10:  Yo ⎬  Reference point (location) of object
       11 - 15:  Zo ⎭  (See Figures)
       16 - 20:  θ  ⎫  Orientation (rotation) of object in degrees.
       21 - 25:  ψ  ⎬  θ = Azimuth about Z - axis: Turn)  From + axis
       26 - 30:  Ø  ⎭  ψ = Pitch about Y - axis: Tilt  )  to minus axis
                       Ø = Roll about X - axis: Aspect )  the angle is +
                                                          when CCW
       31 - 35:  F - scale factor
```

If K = 1

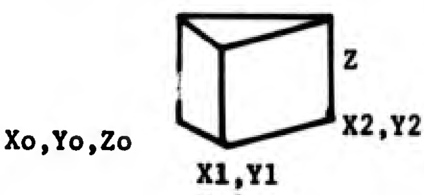
Card D'

```
Columns 1 - 5 :  X1   ⎫ The remaining points in the object base with
        6 - 10:  Y1   ⎪ reference to Xo, Yo, Zo (before scale and
          .      .    ⎪ rotation).  Use as many cards (≤4) as required
          .      .    ⎬ (Columns 1 - 70) to describe the N points
          .      .    ⎪ (7 points per card).
                 Xn-1 ⎪
                 Yn-1 ⎭
                 Z - Height of object
```

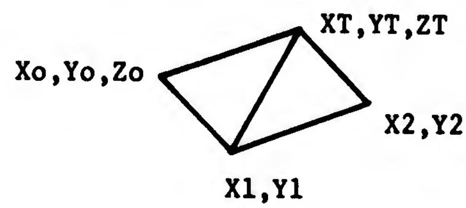
If K = 2

Card D'

Columns 1 - 5 : X_1
6 - 10: Y_1
. .
. .
. .
X_{n-1}
Y_{n-1} } Same as for K = 1
X_t
Y_t
Z_t } Coordinates of top point of object with reference to X_o, Y_o, Z_o



K=1



K=2

If K = 3

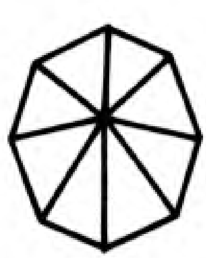
Card D'

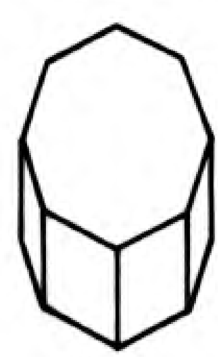
Columns 1 - 5 : R - Radius of circular base of N points centered X_o, Y_o, Z_o. If N (see card C) is blank, 8 points are used.
6 - 10: Z - Height of object

If K = 4

Card D'

Columns 1 - 5 : R - Radius of circular base (See K = 3)
6 - 10: Z - Height of object (above center)


K=4


K=3

c. Repeat

Card C

Columns	1 :	9
	2 - 5 :	N - Object number to be repeated
	9 - 10:	JOBJ non-blank if start of new object

Card D

Columns	1 - 5 :	X_o	Reference point	See b.
	6 - 10:	Y_o	Reference point	See b.
	11 - 15:	Z_o	Reference point	See b.
	16 - 20:	θ	Orientation angles	See b.
	21 - 25:	ψ	Orientation angles	See b.
	26 - 30:	ϕ	Orientation angles	See b.
	31 - 35:	F - scale factor		

No Card D'

All the parts of object N are repeated, further rotated, and further scaled, and placed at X_o, Y_o, Z_o (which corresponds with the first point in the first face in object N).

Card E

Column 5 : (Blank / 2) - End of face data. (Do not / Do) write geometry tables.

The above cards describe the geometry.

Card F

Columns	1 - 5 :	P1 - Factor converting input units to plotter units $\neq$ 0.
	6 - 10:	P2 - Plot reorigin (not used on CRT plotter)
	11 - 20:	ϕ - Elevation angle (degrees) of observer (at origin of figure).
	21 - 30:	E - Area convergence factor (input units), or (if negative) - I - Number of iterations to be run and E is set to .5 (Not used if JR is given). .
	31 - 32:	JR - Number of azimuth inputs (optional) $\leq$ 20.
	33:	IM - Non-blank if only object types change (1 view only)
	34:	I1 - 1, if area calculations are desired.
	35 - 40:	R - Range from origin of figure (optional). Program uses a standard range of 1000.
	41:	S - Symmetry flag { X - X axis; B - Both X and Y Axis }

42: I2 - { 1, if perspective plotting is desired (Iteration 1 only)
2, if perspective and perimeter plotting is desired. }
43: ID1 - Debug flag 1.
44: ID2 - Debug flag 2.
46: ID4 - Debug flag 4.
48 - 53: P3 - Plot reorigin before run (not used on CRT plotter).
56 - 70: INØBJ - Object type (for each of the objects)

IE: Column 56 for Object 1
57 for object 2
. . . .
. . . .
. . . .
70 for object 15

TYPE: 0 - Object is a target (blank column = 0)
1 - Object is a frame
2 - Object is a shield
3 - Object is ignored

A Card F is required for each elevation angle.

Card G (provided if JR is not blank)

Columns 1 - 10: AZM_1
11 - 20: AZM_2
. .
. .
. .
71 - 80: AZM_8

} JR azimuth angles. If JR is greater than 8, additional G cards follow. JR $\leq$ 20.

If this option is used, E and S on card F should be blank.

NOTE: Do not provide a card G if JR (columns 31-32 on card F) is blank.

Card H

Columns 1 - 70: Blank

Jobs can be stacked by inserting additional jobs (cards A through H) here.

Card I

Columns 1 - 70: Blank

Card J

Columns 1 - 70: Blank

In general, input variables starting with letters I through M are right adjusted integers. Otherwise, a decimal point is required, i.e., FC, offsets, X, Y, Z, P1, P2, Ø, E, and P3 require decimals.

4. PROGRAM OUTPUT

The program produces perspective plots of the target under study as well as a perimeter plot. Printed output consists of averaged area of the target from the given elevation angle. This output is in the following form:

AFTER (I4) ITERATIONS, THE AREA = (F8.3)

which gives the iteration number and the area in engineering units.

If additional output is desired, ID1 (card F) may be punched with a 1 or a 2. If ID3 (card A) is a 1, the coefficients of each face are also printed. Other debug options are available as shown in Table I.

TABLE I. DEBUG OUTPUT

VALUE \ ID	1	2	3	4
0	Final Area			
1	θ, AREA (plot) if in trouble, AREA by itera-tion	X, Y projected for each point. FINAL PERIM-ETERS	COEFFICIENT MATRIX and FACE EQUA-TIONS	INTERSECTION POINT table
2	θ,X,Y, AREA (plot) for each view	SEGMENTS of Perimeter		
3		Interaction POINTS by EDGE		
4		SEGMENTS of Perimeter and PERIMETER by PIECE		
5		AREA by segment		
6		"COVER" and "HIDDEN" param-eters for each segment		

If ID contains a value ≥ i, all the outputs for i are given: i.e., if ID2 = 6, all outputs for ID2 are provided.

The program uses the STOPS listed in Table II.

TABLE II. PROGRAM STOPS

STOP	REASON
77	Normal termination
1234	Point number of a face description is too large (card D)
1357	More than 100 points in an object being repeated
2121	More than 25 points on a face (card C)
2442	Error in object number being repeated (card C)
3333	More than 15 objects
4444	More than 900 edges
5555	More than 300 faces
6736	More than 20 points in a solid's base (card C)
6761	More than 600 points (caused by solid)
6767	More than 600 points (card A)
7654	More than 50 segments on one line edge
7766	More than 99 points (198 segments) in perimeter

5. PROGRAM APPLICATION

This program can be used in determining the vulnerability of an aircraft. In this application a generic fighter aircraft was chosen as the target. No shields were used and the outer shell was ignored. First a drawing was made of the vulnerable components of the fighter. The center of the nose tank was chosen as the origin, and the X axis ran the length of the fighter to give one axis of symmetry. (It is assumed that only a direct hit of a vulnerable component will kill the fighter.) Then these components were approximated by use of the components standard solids. (Solid types 2, 3, and 4 were not used in this application.) The complex nose tank could not be approximated by a standard solid; therefore, it was broken into 30 points and 38 faces. (Because some faces were tilted, they had to be broken into several faces to guarantee point planarity within a face.) The remainder of the components consisted of 11 standard solids. The nose tank required 83 cards for its description while only 33 cards were required for the rest of the description. These cards, which are listed in Figure 16, and a card F are used to request a view from an elevation angle of 30 degrees and at an azimuth of 45 degrees. The resulting drawing of this target is shown in Figure 17.

```
   30  1                          1.76
     -.450-.220       -.575 .195        -.975 .345        -.640 .960        -.4201.060
     -.2301.060       -.230 .605         .230 .605         .2301.060         .4201.060
      .640 .960        .975 .345         .575 .195         .450-.220-.10 -.2301.060
-.10  .2301.060-.68 -.1851.195-.68  .1851.195-.94  .1751.170-.94 -.1751.170
-.94 -.615 .960-.94 -.840 .605-.94 -.305 .645-.94 -.160-.180-.94  .160-.180
-.94  .305 .645-.94  .840 .605-.94  .615 .960-.59 -.160-.380-.59  .160-.380
   14     1
  1  2  3  4  5  6  7  8  9 10 11 12 13 14
   10
 19 20 21 22 23 24 25 26 27 28
    3
  6  7 15
    3
  8  9 16
    5
 16  7 15 16  7
    5
 16  7  8 16  7
    5
  5 15  6  5 15
    7
 15  5 17 15  5 15 17
    7
 15 17 18 15 17 15 18
    7
 18 15 16 18 15 18 16
    7
 16 10 18 16 10 16 18
    5
 16 10  9 16 10
    5
 17 19 18 17 19
    5
 17 19 20 17 19
    5
 17 21 20 17 21
    7
 21 17  5 21 17 21  5
    5
 21  5  4 21  5
    5
 21  3  4 21  3
    5
  3 21 22  3 21
    5
 22  2  3 22  2
    5
  2 22 23  2 22
    5
 23  1  2 23  1
    5
  1 23 24  1 23
    3
  1 24 29
    5
 29 14  1 29 14
    5
 14 29 30 14 29
    5
 29 25 24 29 25
```

Figure 16. Input Data for Generic Fighter Aircraft Fuel Tank Target

```
     5
25 29 30 25 29
     3
14 25 30
     5
26 14 25 26 14
     5
26 14 13 26 14
     5
26 12 13 26 12
     5
12 26 27 12 26
     5
28 12 27 28 12
     5
28 12 11 28 12
     5
28 10 11 28 10
     7
28 18 10 28 18 28 10
     5
28 18 19 28 18
1  10     2
-.970      1.210      -90.0        1.0
-.445 .810-1.26 1.03-1.26 .695-.710 .500-.495     -.710-.500-1.26-.695
-1.26-1.03-.445-.810 2.85
1   4     3
-1.89 .980-.240       -90.0        1.0
-.660-.340-.430-.730      -.380 1.34
1   6     4
-3.88      1.210      -90.0        1.0
-.445 .810-.710 .500-.495     -.710-.500-.445-.810 1.40
1   5     5
-3.86      -.780      -90.0        1.0
 .150 .485-.320 .550-.320-.550 .150-.485 1.44
1  10     6
-5.32      1.210      -90.0        1.0
-.445 .810-1.26 1.03-1.26 .695-.710 .500-.495     -.710-.500-1.26-.695
-1.26-1.03-.445-.810 1.06
1  10     7
-6.47      1.210      -90.0        1.0
-.445 .810-1.26 1.03-1.26 .695-.710 .500-.495     -.710-.500-1.26-.695
-1.26-1.03-.445-.810 1.32
9   7     8
-7.86      1.210
1   6     9
-2.41 1.18 .100                    1.0
-1.51      -3.49 1.89-2.83 2.38-2.44 2.36 .31  .71  .295
9   9    10
-2.41-1.18 .395             180.0  1.0
1   4    11
-6.45 1.18 .105                    1.0
-1.37      -1.37 2.33       2.33 .280
9  11    12
-6.45-1.18 .385             180.0  1.0

 1.0  14.       30.0          1 15000.0 22
  45.
```

Figure 16. (Concluded)

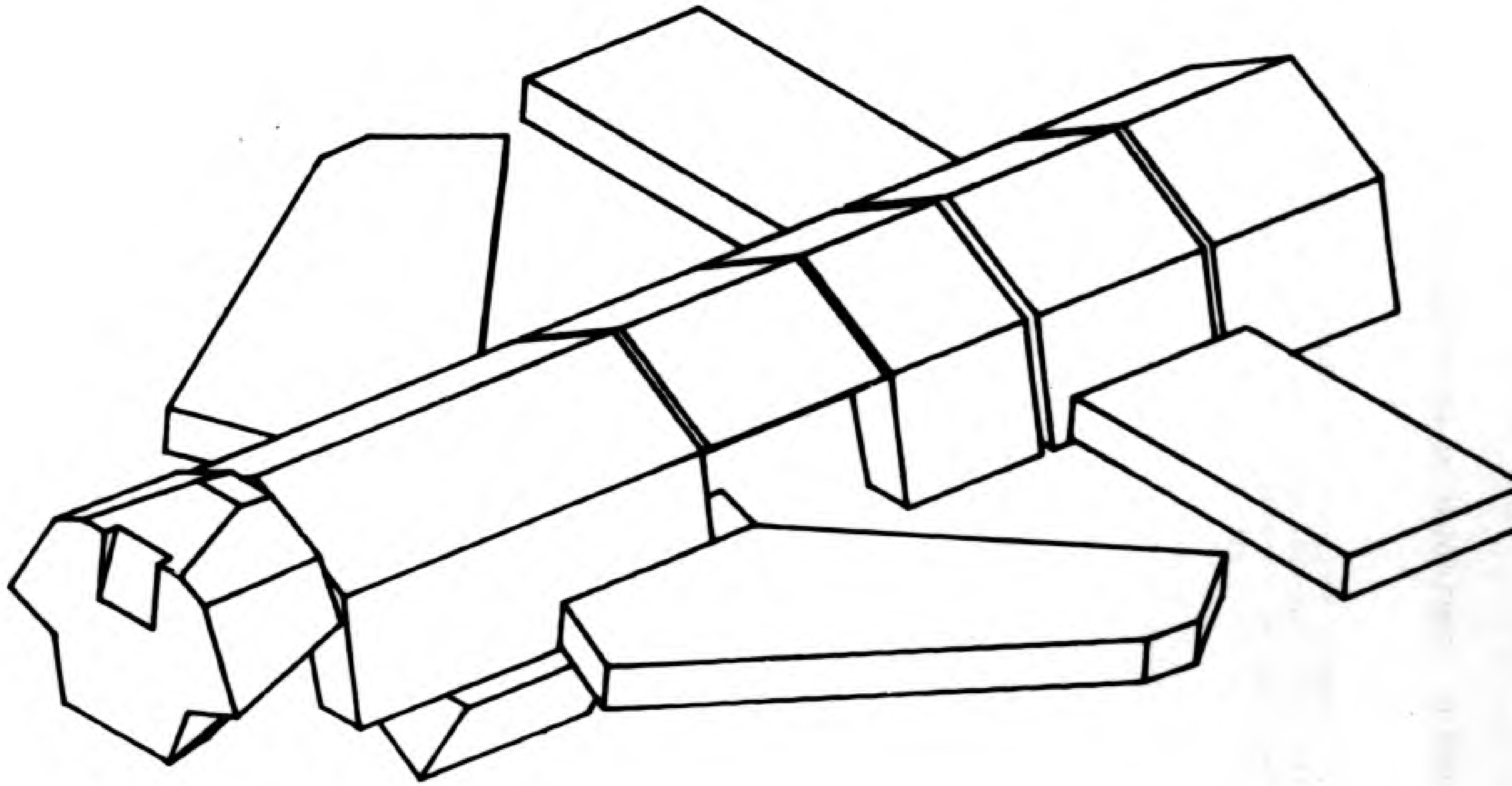


Figure 17. Fuel Tank - Generic Fighter Aircraft

SECTION IV. CONCLUSIONS AND RECOMMENDATIONS

1. CONCLUSIONS

In conclusion, a computer program was developed for rapidly performing target description and vulnerability studies. The program accepts input data in several forms, computes the presented area of a target from any viewing angle, and produces plots with hidden lines removed.

The program is written in modular form to permit future modification with minimal effort. State-of-the-art technology is employed in the target evaluation model.

2. RECOMMENDATIONS

Based on current target evaluation technology and the experience and knowledge acquired during the conduct of this program, it is recommended that the following items be investigated to further develop the capability to perform target evaluations:

1 Develop methods for using curved surfaces in the program.

2 Formulate additional input formats for special target cases.

3 Maintain program in highly efficient form by incorporating new technology into the program as it is developed.

4 Develop methods for combining digital and analog computer routines for time-sequencing control of dynamic target displays.

APPENDIX I

DETAILED INPUT FORMAT (See Section III, 3)

Card A: (I5, 2X, I1, 1X, I1, 4F5.0, I5)

J - The number of points to be read from card B. This value must not be zero (or blank); if all input is in the form of solids (i.e., no point input), J should be set to 1 and a blank card used for card B. This will define the first point as 0., 0., 0. and will never be used, except in the calculation of minimum and maximum point. If a 0., 0., 0. point is not in the range of the solid input, a non-blank card B should be used defining a more realistic dummy first point.

ID3 - If this is a 1, the coefficient matrix and face equations will be printed as they are calculated. This is an extra output option and need not be used unless the face geometry is being debugged. If all faces are defined by solids (rather than by points), this output is not needed.

IØ, ØFFSETX, ØFFSETY, ØFFSETZ - These are normally blank. But if IØ = 1 the offset values will be added to each point on card B.

FC - This factor converting input units to engineering units is only used in the calculation of the final area (i.e., area in input units times FC^2 equals area in engineering units). If FC is blank (or zero), it is set to 1.0.

JDID - This is a special flag for overlay plotting and should not be used with CRT output.

Card B: (15F5.0)

This card(s) defines the J input points as indicated by card A. At least one card must be present.

Xi, Yi, Zi - These are the first J points to be stored in PVXYZ and are in input units. They can be offset by using the appropriate option on Card A, but normally no offset is required. The points need not be in any order, and some may be defined but never used. However, the order the point appears in this list becomes the point's "point number". Columns 76-80 can be used for point number identification (user's aid only). Since there are five points to a card, this I.D. could be 001, 006, 011, etc. to identify the point number of the first point on each card. Blank fields are treated as zero, and (since points can be repeated and not used) points can be deleted from any card without causing any problem (other than wasting a point number).

Card C: (I1, I4, I5)

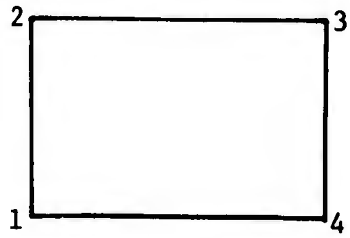
K - Type being described. If this is blank (or zero), a face is being described. If K is less than 5, a solid is being defined. If K=9, an existing object is to be repeated.

N - Number of points in the description (see Card D).

JØBJ - Object number, if different from the object number on the previous card C. The object number of the first card C is assumed to be 1. If the next card C has a blank (or zero) for JØBJ, that description will also be included as part of object 1. Whenever JØBJ is entered (non-zero), it must be exactly one increment greater than the previous object number. (Setting JØBJ to 1 on the second card C is not correct since the first card C is assumed to be object 1. A blank JØBJ on the second card C will include that description as part of object 1.) Only the objects having a unique type code on some card F need be identified; otherwise, all descriptions can be included under one object and JØBJ can be blank on all card C's. Another reason for using object numbers is that repeat cards can be used to repeat an object with similar geometry construction.

Card D (for faces): (2513)

Ni - The N point numbers, IFP, describing the face. The point numbers refer to the order in which the points were defined on card B. The point numbers must be given in either clockwise or counter-clockwise order around the face. All points given must be in a plane or the face should be broken into several triangular faces by means of pseudo edges. For example, to describe the following (tilted) face,



the following could be entered:

Card C: 3

Card D: 1 2 3 1 3

Card C: 3

Card D: 1 4 3 1 3

This double description of edge 1-3 defines a pseudo edge which will not be plotted.

If the face was not tilted (i.e., same Z or Y or X values for each point), a pseudo edge is not required and the following could be entered:

Card C: 4

Card D: 1 2 3 4

NOTE: Since the first three points given on card D define the plane on the face, they must be unique; i.e., the pseudo edge should be defined at the end of the card.

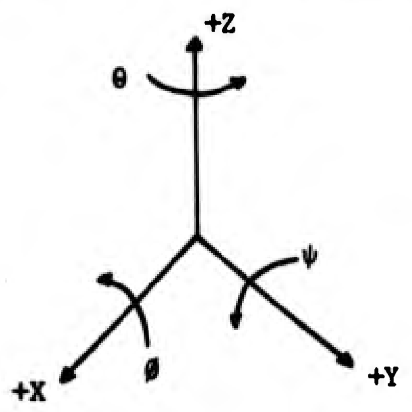
For the tilted face given above, card D should not be: 1 3 1 2 3

Card D (for solids): (7F5.0)

Xo, Yo, Zo - The reference point for this solid. This becomes the location and first point for the solid.

NOTE: All points generated by solid descriptions are included in PVXYZ, the point array initiated by card B, and so are included in point limitation count.

θ, ψ, ∅ - The orientation of the solid in degrees.



θ = Azimuth or turn

ψ = Pitch or tilt

∅ = Roll or aspect

When looking down an axis from the plus side to the minus side, the corresponding angle is plus when counterclockwise and negative when clockwise.

F - This scale factor (FCT) is used only if the coordinates given on the following card(s) are not in input units; then each coordinate will be multiplied by this factor. If F is zero (or blank), it is set to 1.0.

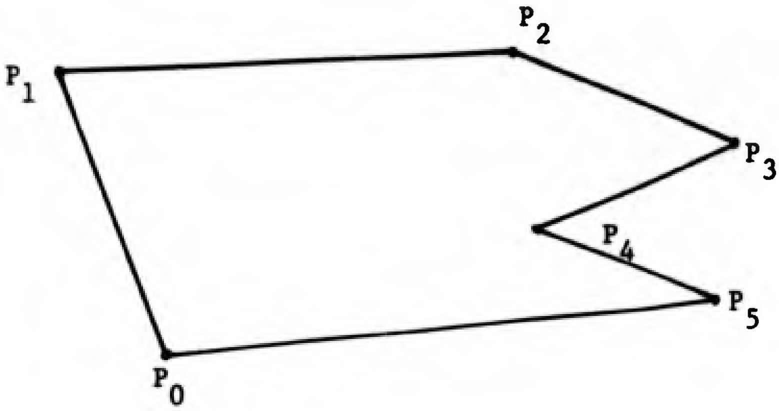
NOTE: All units are in input units unless otherwise denoted.

The type of solid (K, on card C) being described is determined as follows:

K = 1: Right polyhedron

This normally is the most common solid used. If the base of the solid is congruent to the top, and the lines connecting the common points form rectangles, the solid is a right polyhedron.

The base is initially described as being the X-Y plane, with translation, scaling, and rotation being accomplished by the information given on card D. There may be as many points (within reason) in the base as desired, one of which is the reference point.

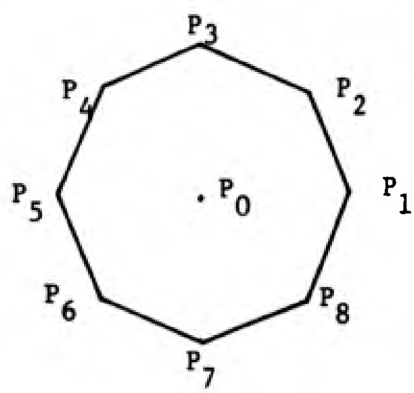


K = 2: Pyramid

This solid has a similar base to that described above, but the top is a single point instead of a plane.

K = 3: Right Cylinder

This type of solid is a right polyhedron whose base is (an approximation to) a circle of N (see card C) points centered about the reference point. N need not be given; if it is not given, the circle will contain eight points. Note that the reference point for this solid is the center of the (circular) base. As is the case for the right polyhedron, the top of the solid is assumed congruent to the base and is in the X-Y plane until translated (by the reference point), scaled (by the factor), and rotated (by the orientation).



K = 4: Right Cone

This solid has a similar base to that described above, but its top is a single point (right) above the base center.

Card D': (14F5.0)

Xi, Yi - The N-1 points needed to complete the base of the solid (in the X-Y plane). The points must be given in either clockwise or counterclockwise order starting from (but not reentering) the reference point. The reference point is assumed to be at 0, 0, 0.

Z (not used for type 2) - This is the height of the solid and is the distance between the solid's base and top.

XT, YT, ZT (Type 2 only) - This is the solid's vertex and is given in three-space with respect to the reference point (i.e., assuming the reference point is 0, 0, 0).

R (types 3 and 4) - This is the radius of the (circular) base.

Repeat Card

If K (on card C) is a 9, a repetition of a previously defined object is accomplished.

N is the number of the object to be repeated. This number is determined by JØBJ on previous card C's. The object may consist of one face, many faces, several solids, or any other combination of the above types. The reference point of an object is the first point in the object. If the object contains only solids, this first point is the reference point of the first solid, defined in that object (i.e., the one with a nonblank JØBJ).

JØBJ on the repeat card determines whether this new (repeat) description starts a new object definition. If not, this description becomes a part of the object being defined on the previous card C.

A card D is used following a repeat card to define the translation, rotation, and scale of the new description. If the original object had parts that had been rotated or scaled, they will be further rotated about the object's reference point (see above definition) and further scaled.

Card E: This is actually a card C that has N less than 3. Since no face can have less than three points for its definition, this card terminates the geometry input section.

If N is a 2, the geometry tables that had been generated are printed out. (Appendix IV contains a description of these tables.)

Card F: (2F5.2, 2F10.3, I2,2I1, F6.2, A1, 5I1, 1X, F6.2, 2X, 15I1)

P1 - This plot scale factor is used for plotting purposes only and will not affect the final printed area. P1 should be chosen so that, when an input value is multiplied by P1, the result is less than 8. In fact, the difference between the maximum scale value and the minimum scale value should be less than 8. P1 is used to convert input units to inches. P1 should not be blank. If no scaling is needed, 1 should be entered.

Ø - This is the elevation angle of the line of sight to the target's origin measured from the X-Y plane of the target. It is entered in degrees.

E - This is the area convergence factor in input units, if an iterative run is being made. If the average target area seen after i iterations is within E of the previous average, then the run is considered to have converged. The program will terminate the run after six iterations in any case. Another method of running is to indicate the maximum number of iterations to be run. A negative number in the E field will indicate this option (i.e., if -2. is entered for E, two iterations will be run). In this alternate mode, the program will also terminate the run if the target areas after two consecutive iterations differ by less than .5. If the JR option is being used, E is ignored.

JR - This is the number of azimuth angles entered on the next card (card G) to view the target from particular azimuths (instead of iterating).

IM - This flag can be used to save a computer run time. If the only change on this card F from the previous card F is in columns 56-70 (object type), a 1 can be punched in this column). Note: This can only be used if JR is 1.

I1 - This flag indicates whether or not the area of the target is to be calculated. If left blank, the most that should be expected from the run is a perspective plot.

R - Range from the target's origin in input units. Normally, this is blank and the program will use 1000. If some smaller range is desired, it may be entered. (Larger values of R will not appreciably affect the results.)

S - If the geometry is set up so that the target is symmetric along the X axis (the X axis running through the center of the target), the character X can be punched. If the target is symmetric along both axes, the character B can be punched. This symmetry flag is only used to reduce the number of views needed in an iteration to calculate the average area.

I2 - This flag indicates whether plotting is desired. A 1 will give perspective plots, and a 2 will give both perspective and perimeter plots. If an iterative run is being made, plots will be produced during iteration 1 only.

ID1, ID2, ID4 - Debug output flags. See Section III, 4. Normally, ID1 is a 1 or 2 while ID2 and ID4 are blank.

INØBJ - This defines the types for the objects in the run. A blank or zero column indicates that the corresponding object is part of the target. A 2 indicates the object is a shield; a 1 indicates that the object is just a frame. If an object is to be ignored completely during this run, a 3 should be punched in the corresponding column.

Card G: (8F10.4)

AZM - These are the azimuths at which a view is to be made. If JR on card F is blank, this card must not be entered. Otherwise, views will be taken from the first JR fields on the card.

Card H:

This is a card F with P1 blank, and will terminate the job. Jobs may be stacked and the program will terminate when two additional blank cards follow a card H (also blank).

NUMBER OF VIEWS

ITERATION NUMBER	SYMMETRY		
	None	X	B
1	4	3	2
2	4	2	1
3	8	4	2
4	16	8	4
5	32	16	8

ie, For two iterations with no symmetry, a total of eight views is obtained.

NOTE: Plots are obtained during iteration 1 only.

ITERATION NUMBER	VIEWING ANGLE, θ, (DEGREES) AZIMUTH (All θ's are in .eased by .1 radians*) (NO SYMMETRY)
1	0, 90, 180, 270
2	45, 135, 225, 315
3	22.5, 67.5, 112.5, 157.5, 202.5, 247.5, 292.5, 337.5
	etc.

*To possibly avoid parallelism problems.

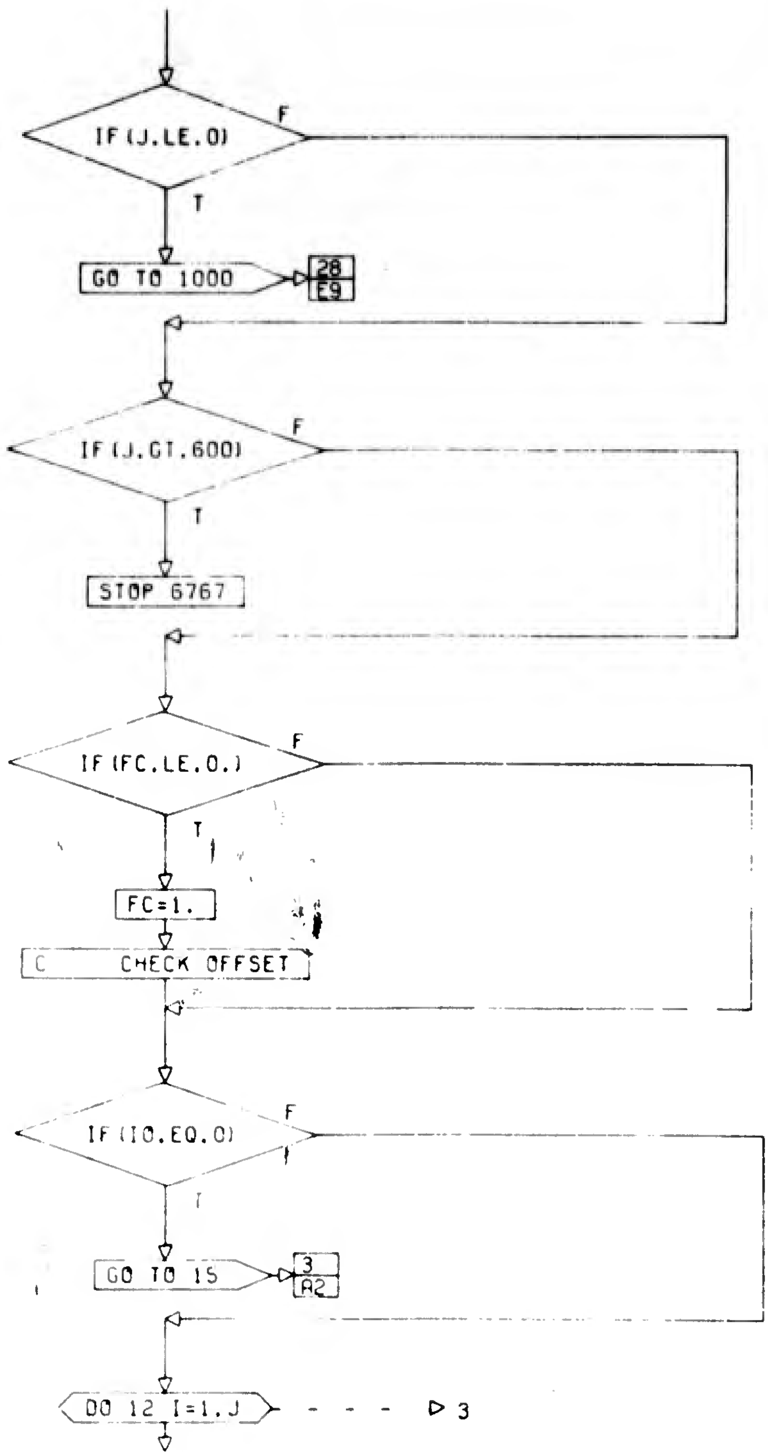
(The reverse of this page is blank)

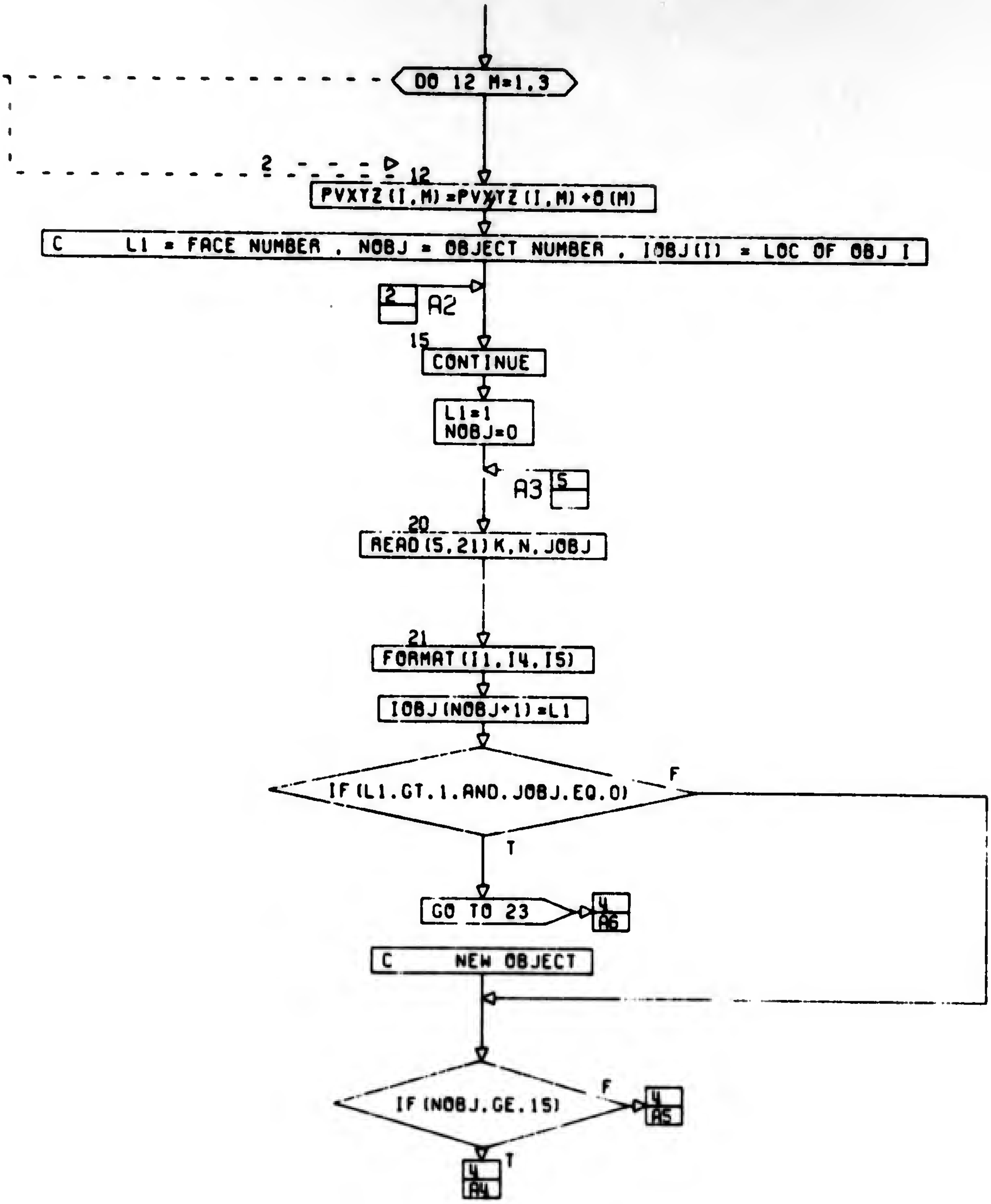
APPENDIX II

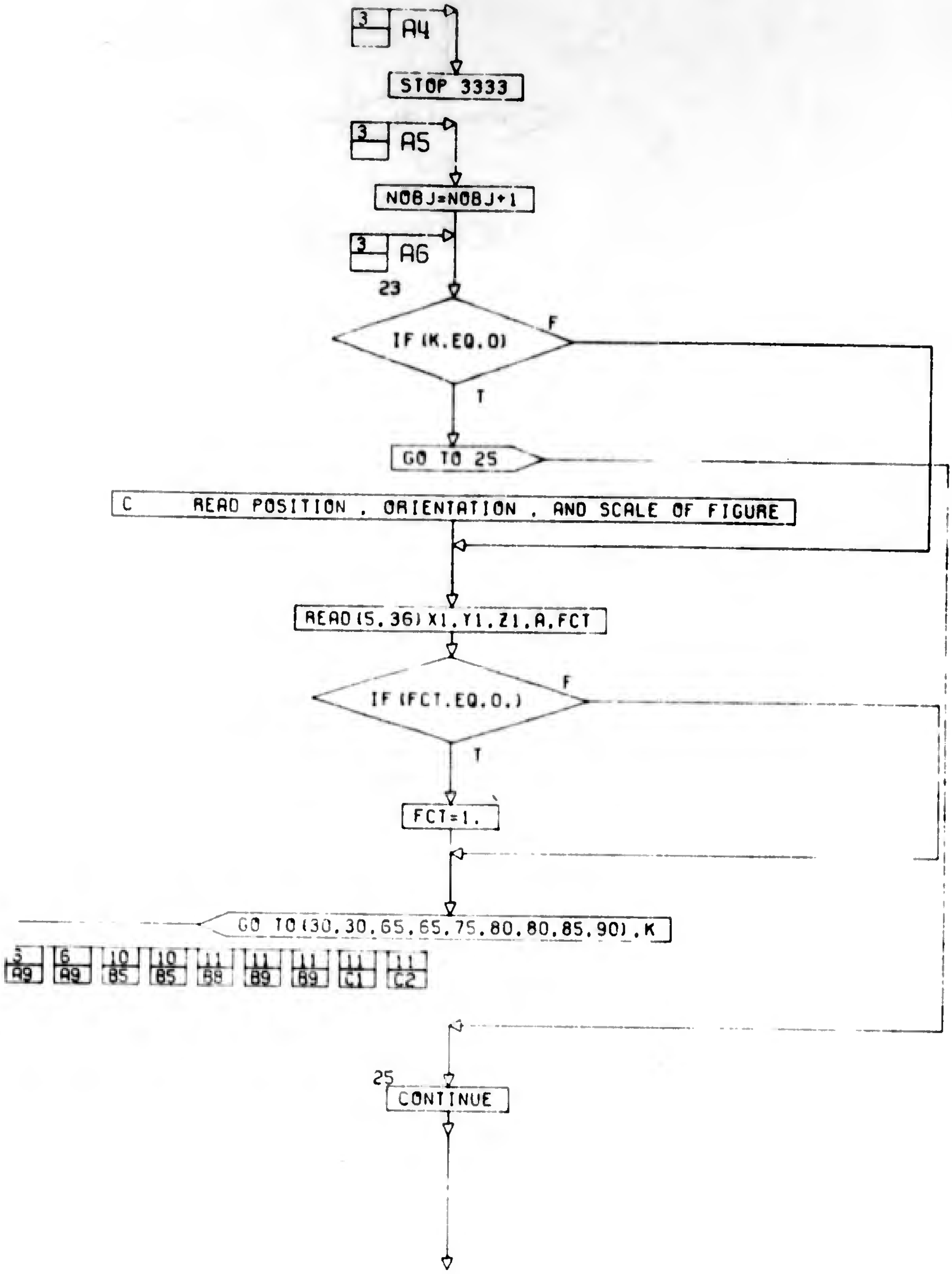
DETAILED FLOW CHARTS

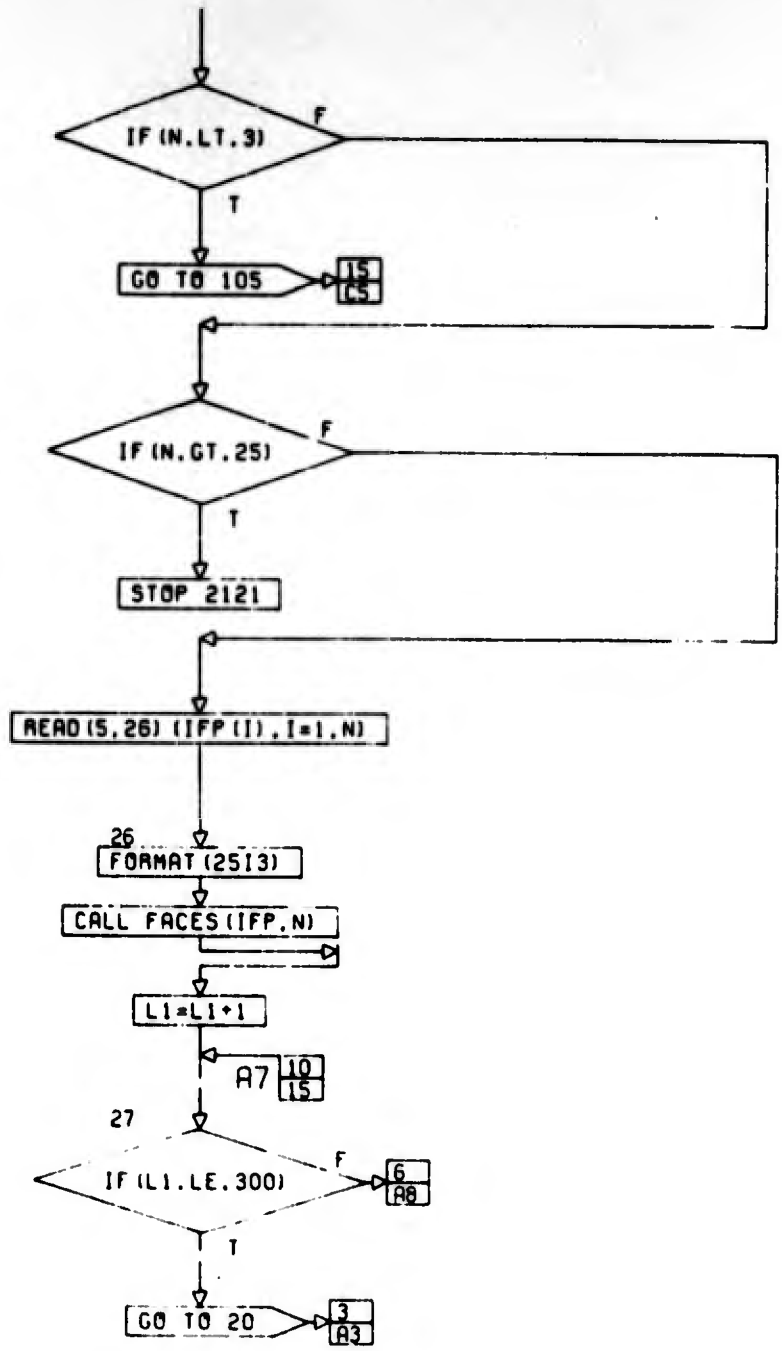
PROGRAM PPLOT

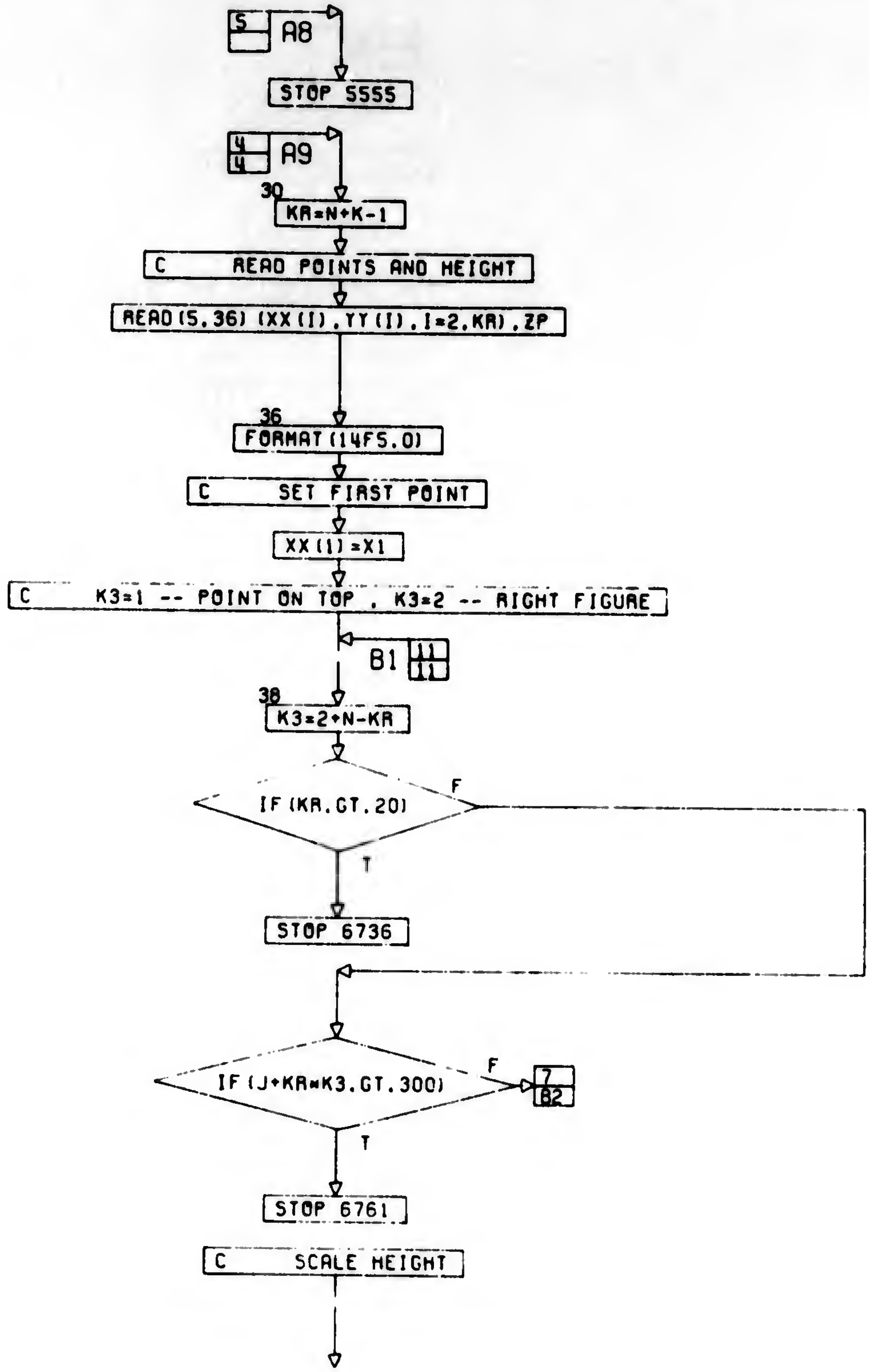
```
C     PROGRAM PPLOT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)

      COMMON/RAST/NOO

      COMMON COEF(300,4)
     .,LFCE(301),JFCE(1800),PVRT(600,2),PVXYZ(600,3),JEDGE(900,5),
      PINT(1200,4),PEYE(3),NMBI(900)
     .,IOBJ(16),INOBJ(15),LFCI(300)

      COMMON /INPTT/J,NE,LL,L,NI,NP,NOBJ,IM
      COMMON/DEBUGS/IBUG(4)
      COMMON/ARCH/PPRP(99,2),NPP
      COMMON/AR/SAREAS(25),NAREAS(25)
      COMMON/RNGE/PMX(3),PMN(3),IDID,JDID
      COMMON/TEMPY/JFCET(1800),IPNT(100)
      COMMON/ROTT3/COSA(3),SINA(3)
      DIMENSION          IFP(25),O(3),SYM(2),AZM(20)

      DIMENSION XX(20),YY(20),ZZ(20),A(3)
      EQUIVALENCE(X,PEYE(1)),(Y,PEYE(2)),(Z,PEYE(3))
      DATA SYM/1HB,1HX/
      DATA PI2/6.283185/

      CALL PLOTCC(0.,0.,-3)

      NOO=100

      A1 18

    5 IDID=0

      WRITE(6,1)

    1 FORMAT(27H1TARGET DESCRIPTION PROGRAM)

C     J=NUMBER OF POINTS

      READ(5,11)J,IBUG,IO,O,FC,JDID,((PVXYZ(I,M),M=1,3),I=1,J)

   11 FORMAT(I5,5I1,4F5.0,I5/(15F5.0))
```

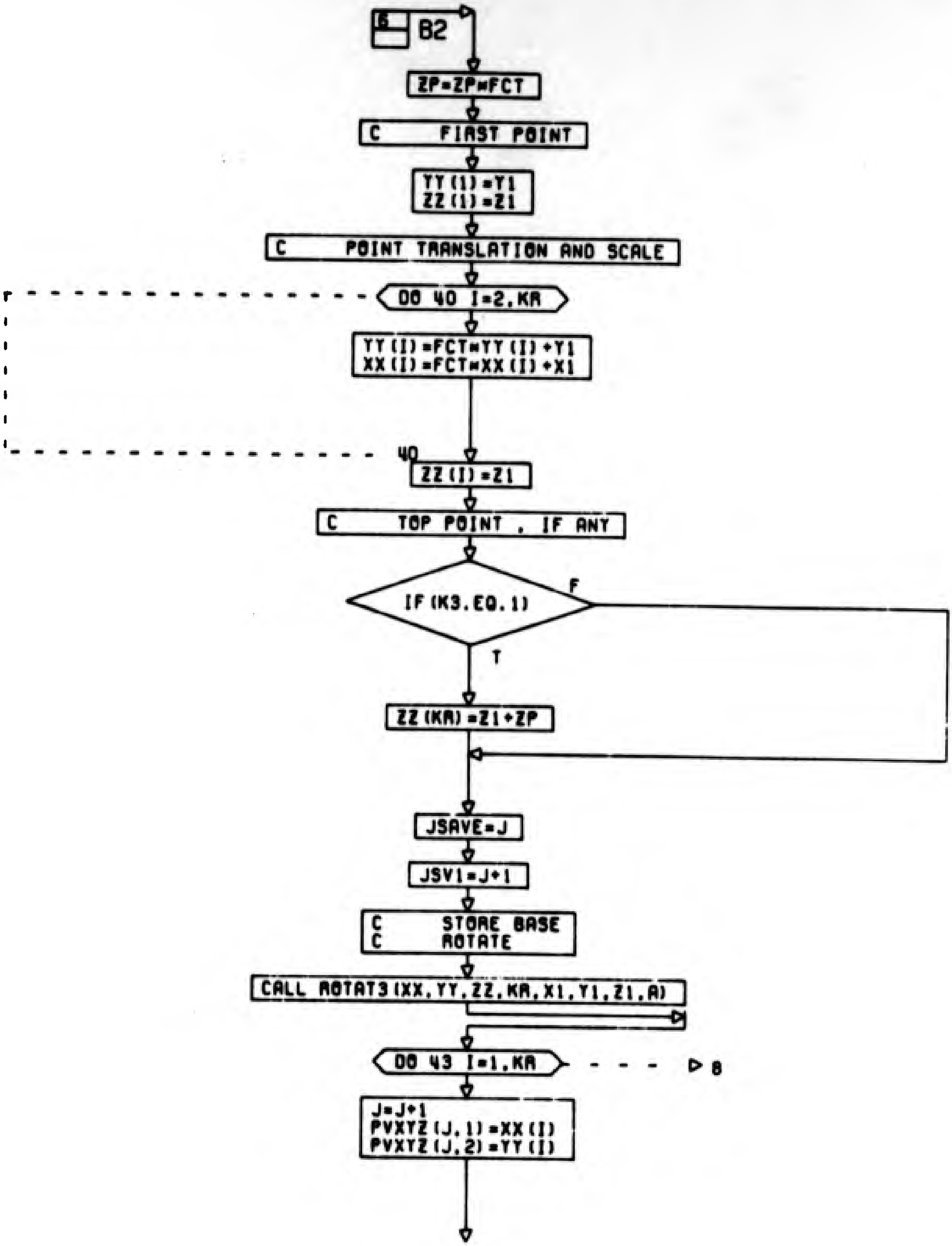

IF (J.LE.0)
F
T
GO TO 1000
28
E9
IF (J.GT.600)
F
T
STOP 6767
IF (FC.LE.0.)
F
T
FC=1.
C CHECK OFFSET
IF (IO.EQ.0)
F
T
GO TO 15
3
A2
DO 12 I=1,J
3

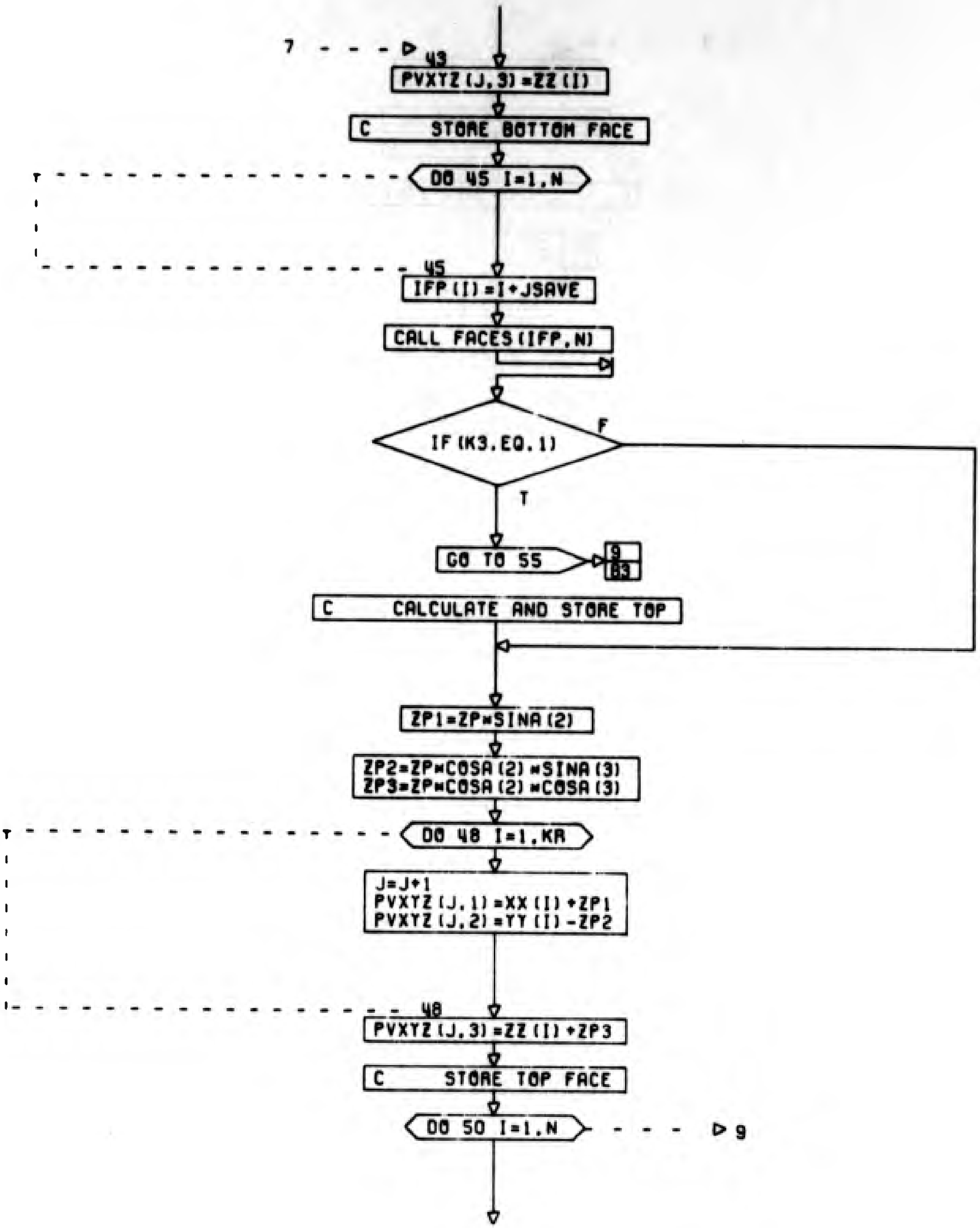

DO 12 M=1,3
2
12
PVXYZ(I,M)=PVXYZ(I,M)+O(M)
C L1 = FACE NUMBER , NOBJ = OBJECT NUMBER , IOBJ(I) = LOC OF OBJ I
2
A2
15
CONTINUE
L1=1
NOBJ=0
A3
5
20
READ(5,21)K,N,JOBJ
21
FORMAT(I1,I4,I5)
IOBJ(NOBJ+1)=L1
IF(L1.GT.1.AND.JOBJ.EQ.0)
F
T
GO TO 23
4
A6
C NEW OBJECT
IF(NOBJ.GE.15)
F
4
A5
T
4
A4

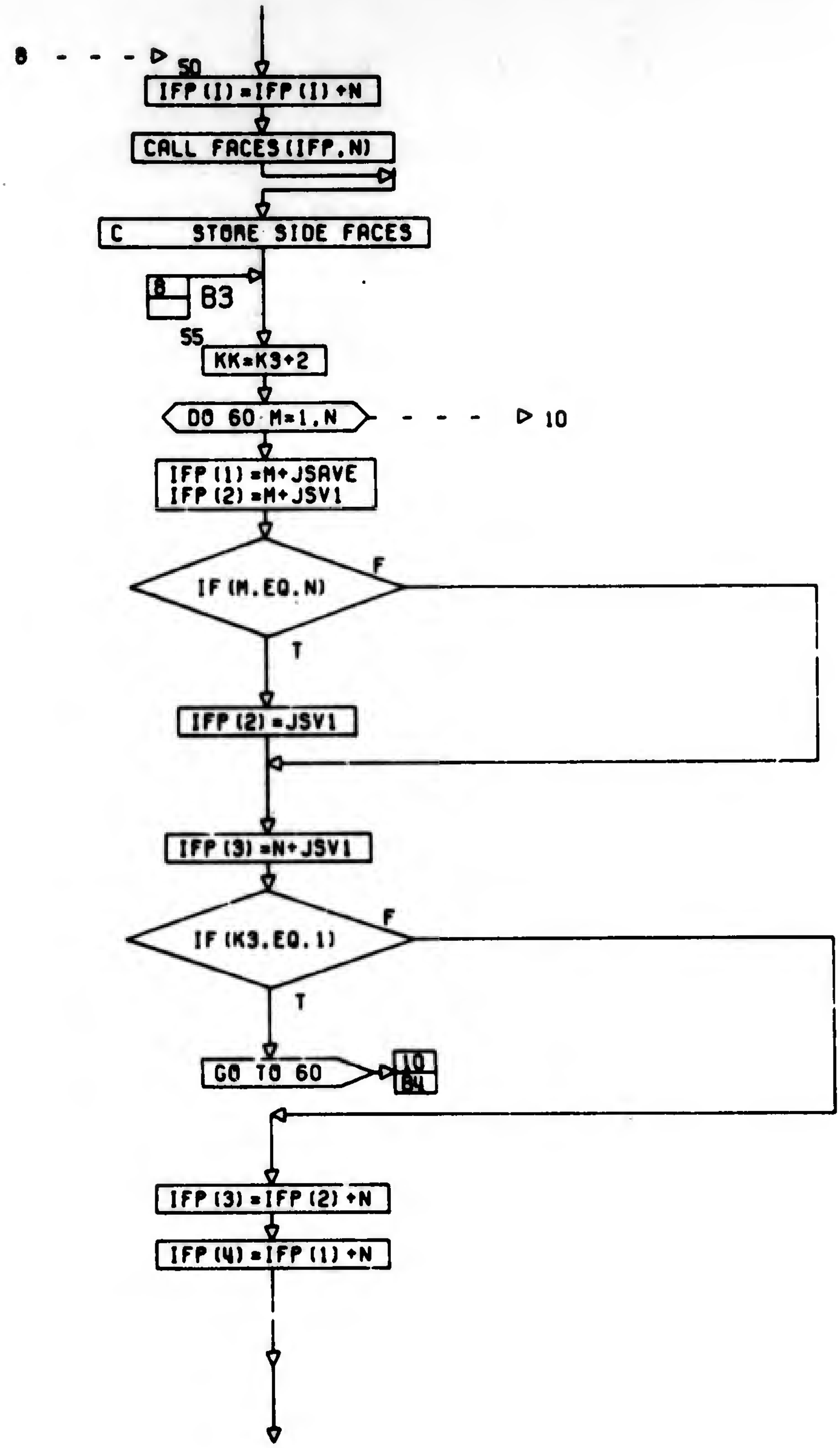

3 A4
STOP 3333
3 A5
NOBJ=NOBJ+1
3 A6
23
IF (K.EQ.0)
F
T
GO TO 25
C READ POSITION , ORIENTATION , AND SCALE OF FIGURE
READ(5,36) X1,Y1,Z1,A,FCT
IF (FCT.EQ.0.)
F
T
FCT=1.
GO TO (30,30,65,65,75,80,80,85,90),K
3 A9
6 A9
10 B5
10 B5
11 B8
11 B9
11 B9
11 C1
11 C2
25
CONTINUE


IF (N.LT.3)
F
T
GO TO 105
15
C5
IF (N.GT.25)
F
T
STOP 2121
READ (5,26) (IFP(I),I=1,N)
26
FORMAT (25I3)
CALL FACES(IFP,N)
L1=L1+1
A7
10
15
27
IF (L1.LE.300)
F
6
A8
T
GO TO 20
3
A3

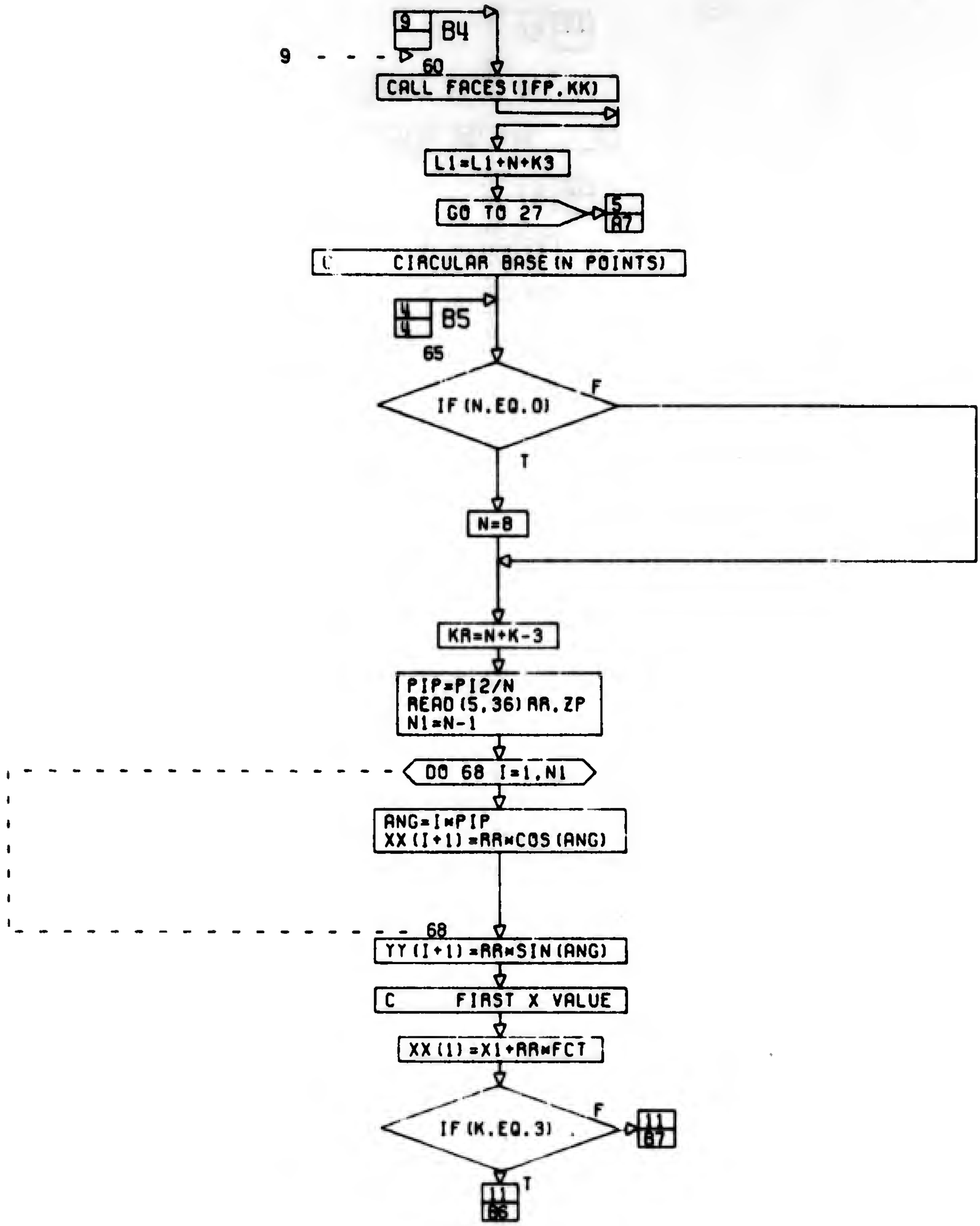

5
A8
STOP 5555
4
4
A9
30
KR=N+K-1
C READ POINTS AND HEIGHT
READ(5,36) (XX(I),YY(I),I=2,KR),ZP
36
FORMAT(14F5.0)
C SET FIRST POINT
XX(1)=X1
C K3=1 -- POINT ON TOP , K3=2 -- RIGHT FIGURE
B1
11
11
38
K3=2+N-KR
IF(KR.GT.20)
F
T
STOP 6736
IF(J+KR*K3.GT.300)
F
7
B2
T
STOP 6761
C SCALE HEIGHT

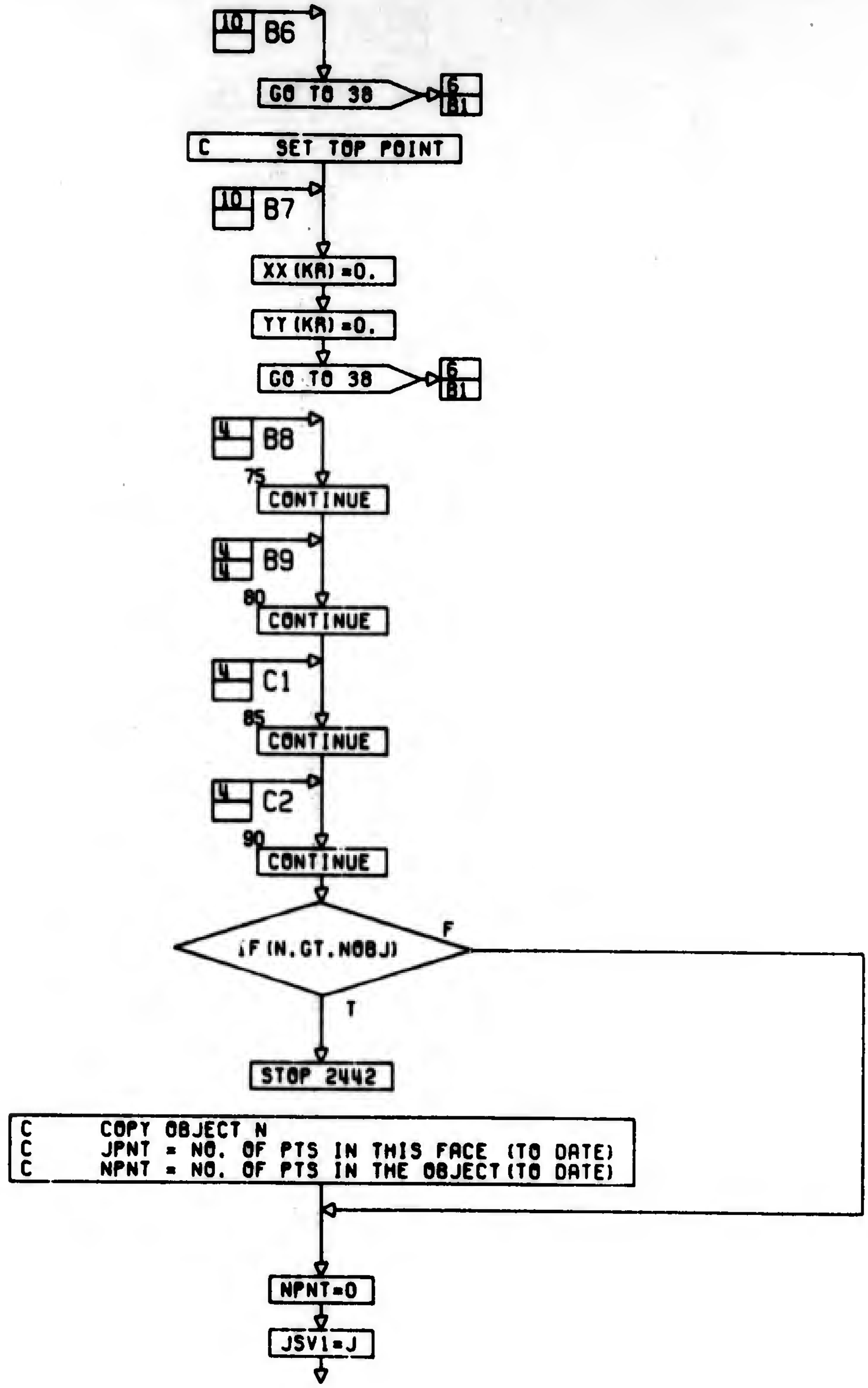

B2
ZP=ZP*FCT
C FIRST POINT
YY(1)=Y1
ZZ(1)=Z1
C POINT TRANSLATION AND SCALE
DO 40 I=2,KR
YY(I)=FCT*YY(I)+Y1
XX(I)=FCT*XX(I)+X1
40
ZZ(I)=Z1
C TOP POINT, IF ANY
IF(K3.EQ.1)
F
T
ZZ(KR)=Z1+ZP
JSAVE=J
JSV1=J+1
C STORE BASE
C ROTATE
CALL ROTAT3(XX,YY,ZZ,KR,X1,Y1,Z1,A)
DO 43 I=1,KR
8
J=J+1
PVXYZ(J,1)=XX(I)
PVXYZ(J,2)=YY(I)

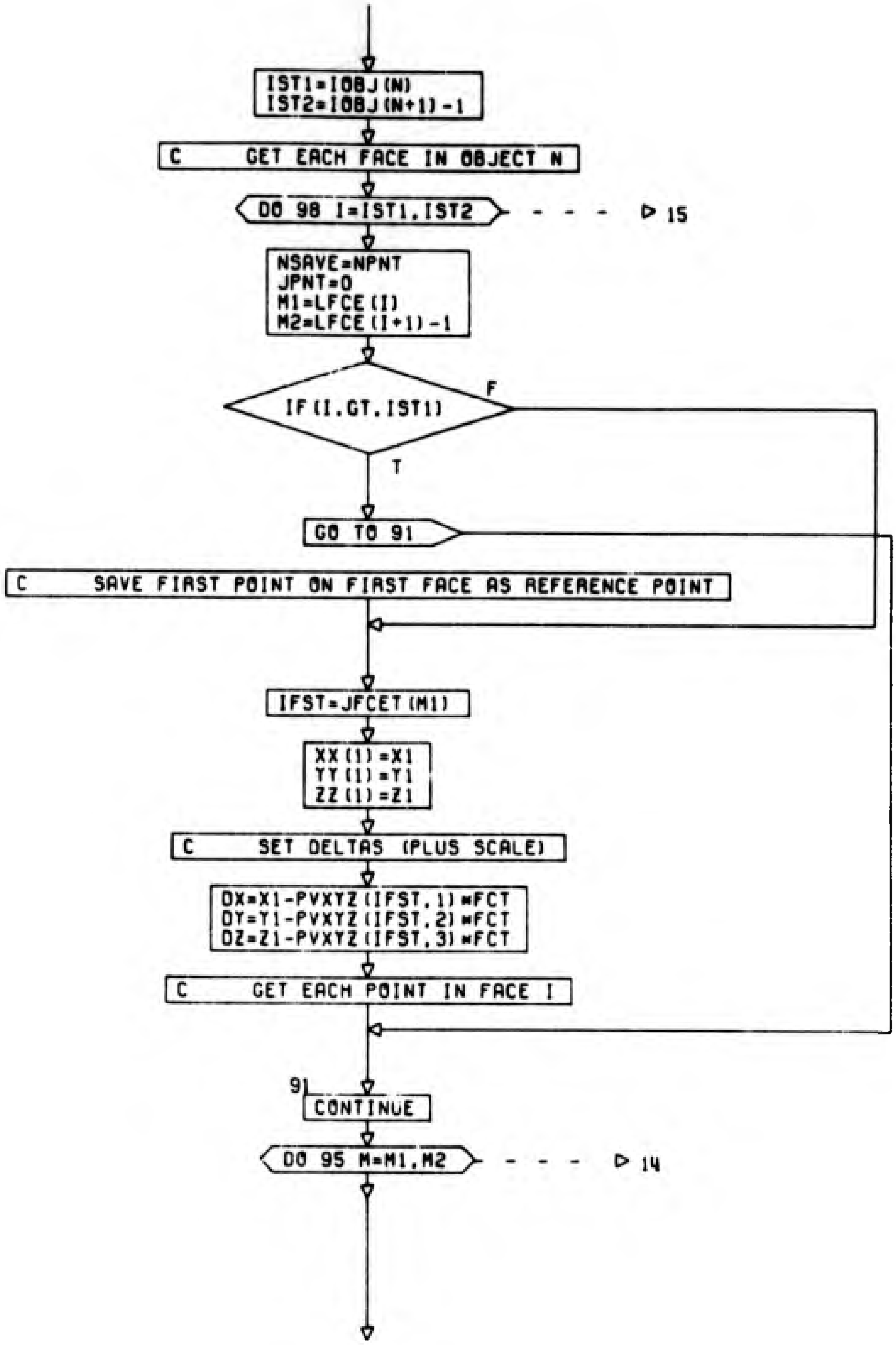

7
43
PVXYZ (J,3) =ZZ (I)
C STORE BOTTOM FACE
DO 45 I=1,N
45
IFP (I) =I+JSAVE
CALL FACES (IFP,N)
IF (K3.EQ.1)
F
T
GO TO 55
9
B3
C CALCULATE AND STORE TOP
ZP1=ZP*SINA (2)
ZP2=ZP*COSA (2) *SINA (3)
ZP3=ZP*COSA (2) *COSA (3)
DO 48 I=1,KR
J=J+1
PVXYZ (J,1) =XX (I) +ZP1
PVXYZ (J,2) =YY (I) -ZP2
48
PVXYZ (J,3) =ZZ (I) +ZP3
C STORE TOP FACE
DO 50 I=1,N
9

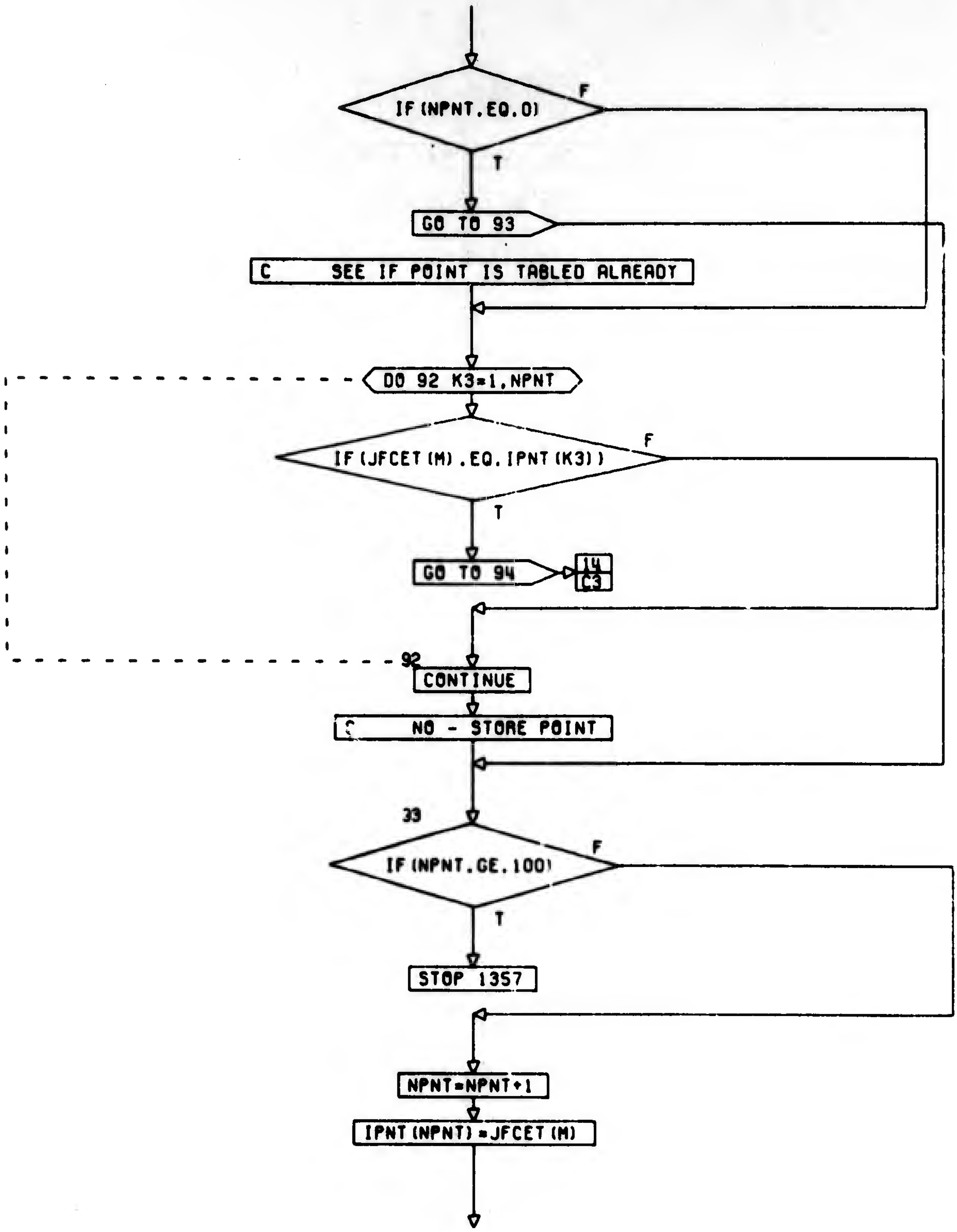


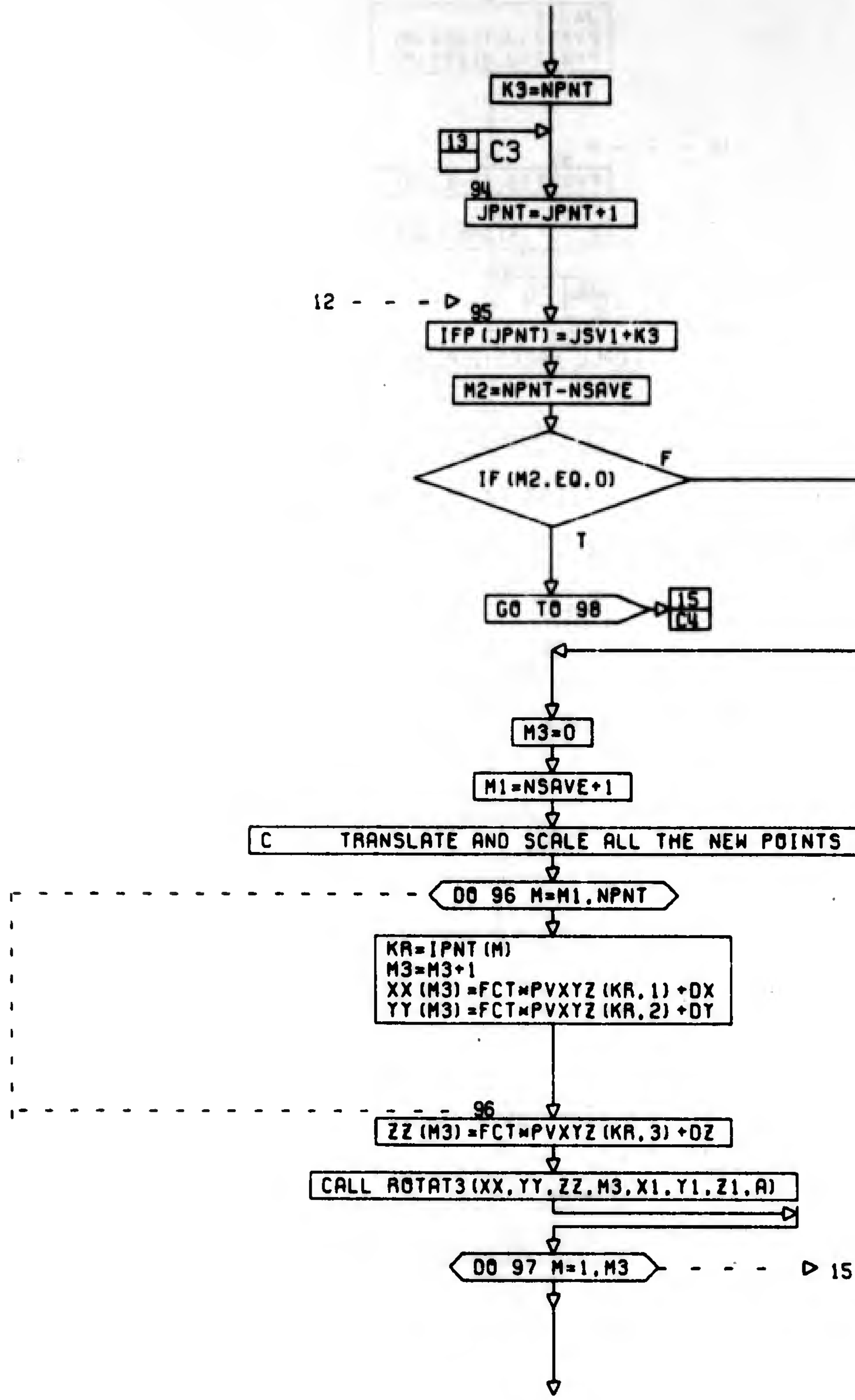
8
50
IFP(I)=IFP(I)+N
CALL FACES(IFP,N)
C STORE SIDE FACES
8
B3
55
KK=K3+2
DO 60 M=1,N
10
IFP(1)=M+JSAVE
IFP(2)=M+JSV1
IF(M.EQ.N)
F
T
IFP(2)=JSV1
IFP(3)=N+JSV1
IF(K3.EQ.1)
F
T
GO TO 60
10
B4
IFP(3)=IFP(2)+N
IFP(4)=IFP(1)+N

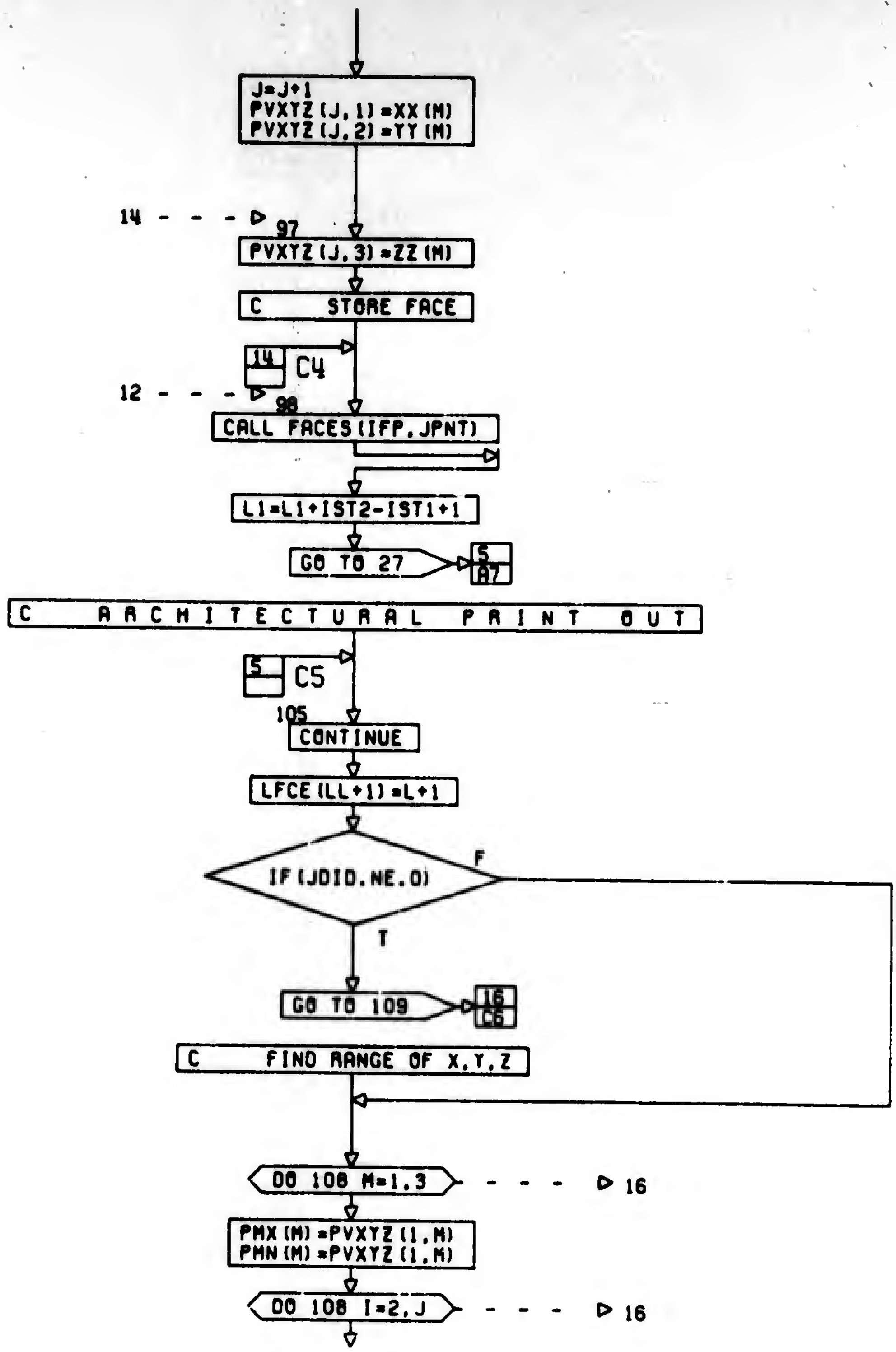

9
B4
9
60
CALL FACES(IFP,KK)
L1=L1+N+K3
GO TO 27
5
A7
C CIRCULAR BASE(N POINTS)
4
4
B5
65
IF(N.EQ.0)
F
T
N=8
KA=N+K-3
PIP=PI2/N
READ(5,36)RA,ZP
N1=N-1
DO 68 I=1,N1
ANG=I*PIP
XX(I+1)=RA*COS(ANG)
68
YY(I+1)=RA*SIN(ANG)
C FIRST X VALUE
XX(1)=X1+RA*FCT
IF(K.EQ.3)
F
11
B7
T
11
B6


10
B6
GO TO 38
6
B1
C SET TOP POINT
10
B7
XX (KR) =0.
YY (KR) =0.
GO TO 38
6
B1
4
B8
75
CONTINUE
4
4
B9
80
CONTINUE
4
C1
85
CONTINUE
4
C2
90
CONTINUE
IF (N.GT.NOBJ)
F
T
STOP 2442
C COPY OBJECT N
C JPNT = NO. OF PTS IN THIS FACE (TO DATE)
C NPNT = NO. OF PTS IN THE OBJECT (TO DATE)
NPNT=0
JSV1=J

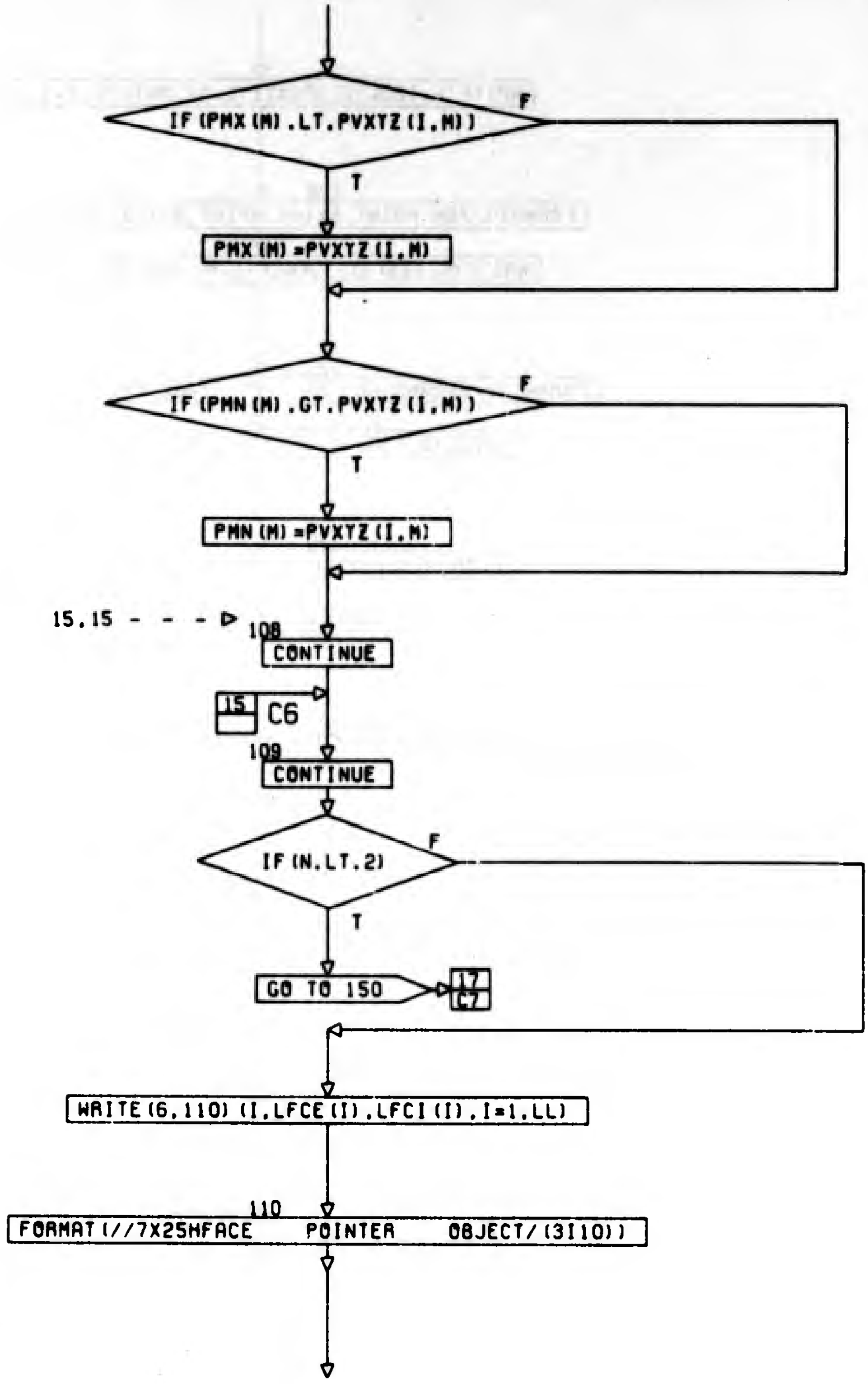

IST1=IOBJ(N)
IST2=IOBJ(N+1)-1
C GET EACH FACE IN OBJECT N
DO 98 I=IST1,IST2
15
NSAVE=NPNT
JPNT=0
M1=LFCE(I)
M2=LFCE(I+1)-1
IF(I.GT.IST1)
F
T
GO TO 91
C SAVE FIRST POINT ON FIRST FACE AS REFERENCE POINT
IFST=JFCET(M1)
XX(1)=X1
YY(1)=Y1
ZZ(1)=Z1
C SET DELTAS (PLUS SCALE)
DX=X1-PVXYZ(IFST,1)*FCT
DY=Y1-PVXYZ(IFST,2)*FCT
DZ=Z1-PVXYZ(IFST,3)*FCT
C GET EACH POINT IN FACE I
91
CONTINUE
DO 95 M=M1,M2
14

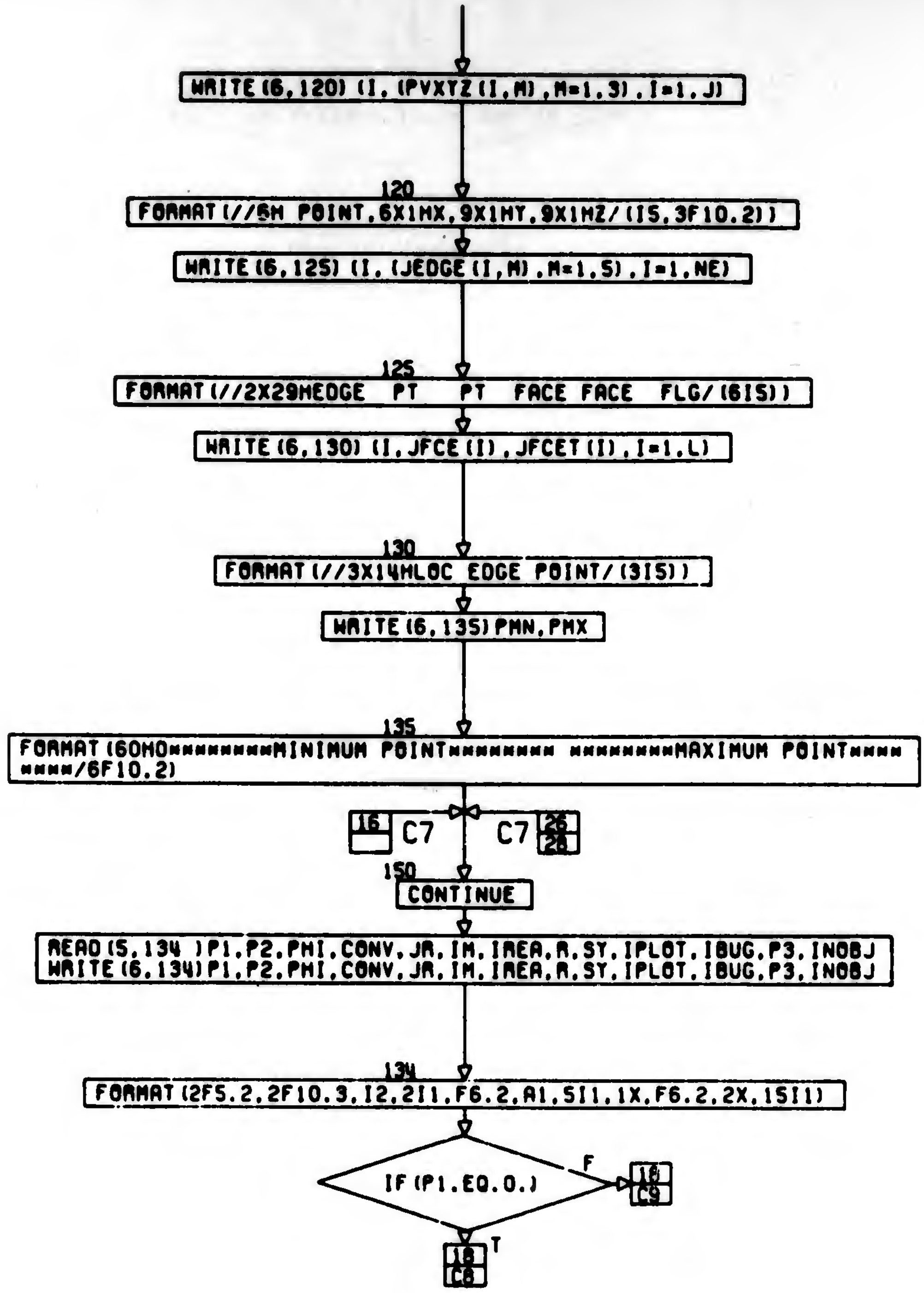

IF (NPNT.EQ.0)
F
T
GO TO 93
C SEE IF POINT IS TABLED ALREADY
DO 92 K3=1,NPNT
IF (JFCET(M).EQ.IPNT(K3))
F
T
GO TO 94
14
C3
92
CONTINUE
C NO - STORE POINT
93
IF (NPNT.GE.100)
F
T
STOP 1357
NPNT=NPNT+1
IPNT(NPNT)=JFCET(M)

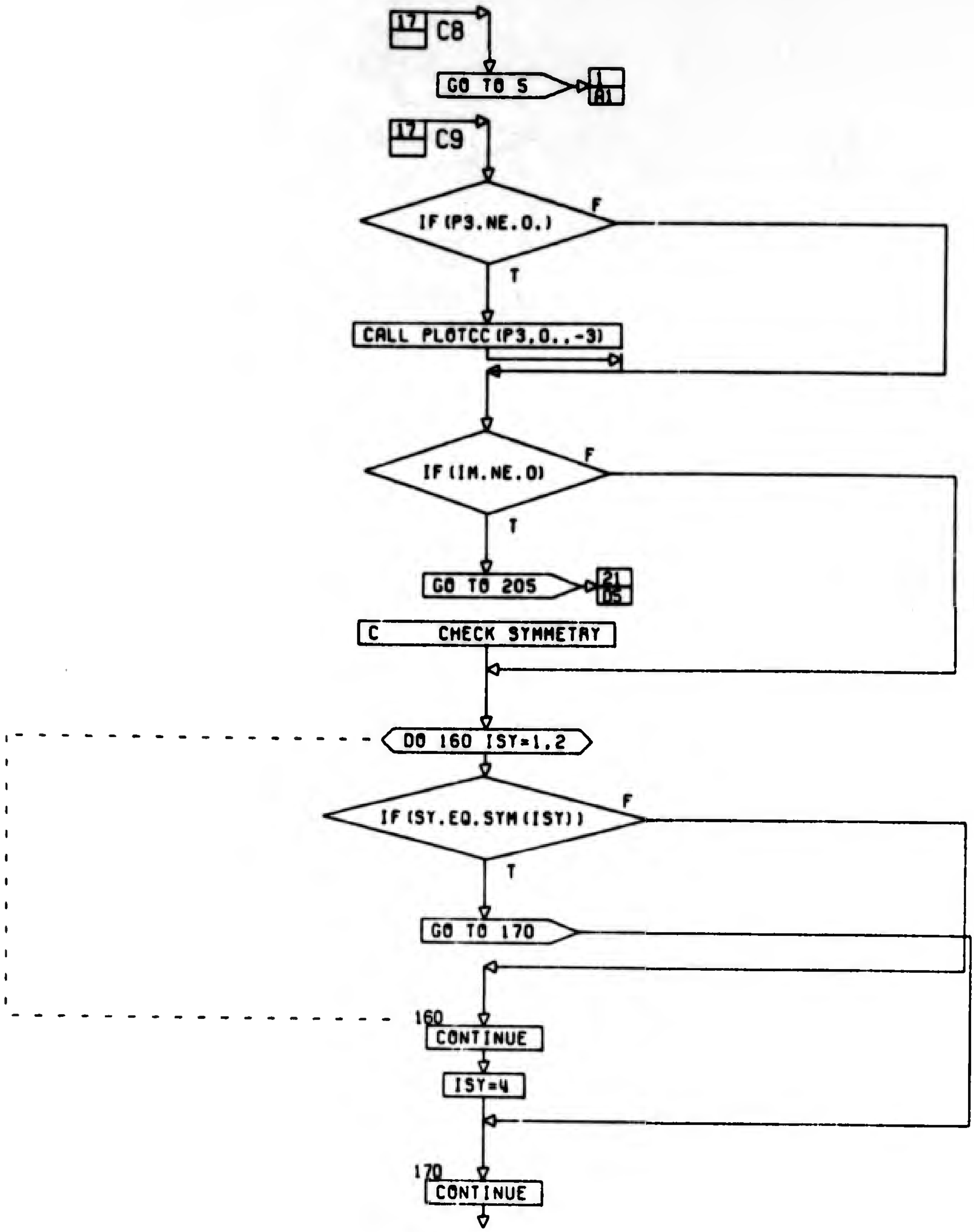

K3=NPNT
13
C3
94
JPNT=JPNT+1
12
95
IFP(JPNT)=JSV1+K3
M2=NPNT-NSAVE
IF(M2.EQ.0)
F
T
GO TO 98
15
C4
M3=0
M1=NSAVE+1
C TRANSLATE AND SCALE ALL THE NEW POINTS
DO 96 M=M1,NPNT
KR=IPNT(M)
M3=M3+1
XX(M3)=FCT*PVXYZ(KR,1)+DX
YY(M3)=FCT*PVXYZ(KR,2)+DY
96
ZZ(M3)=FCT*PVXYZ(KR,3)+DZ
CALL ROTAT3(XX,YY,ZZ,M3,X1,Y1,Z1,A)
DO 97 M=1,M3
15

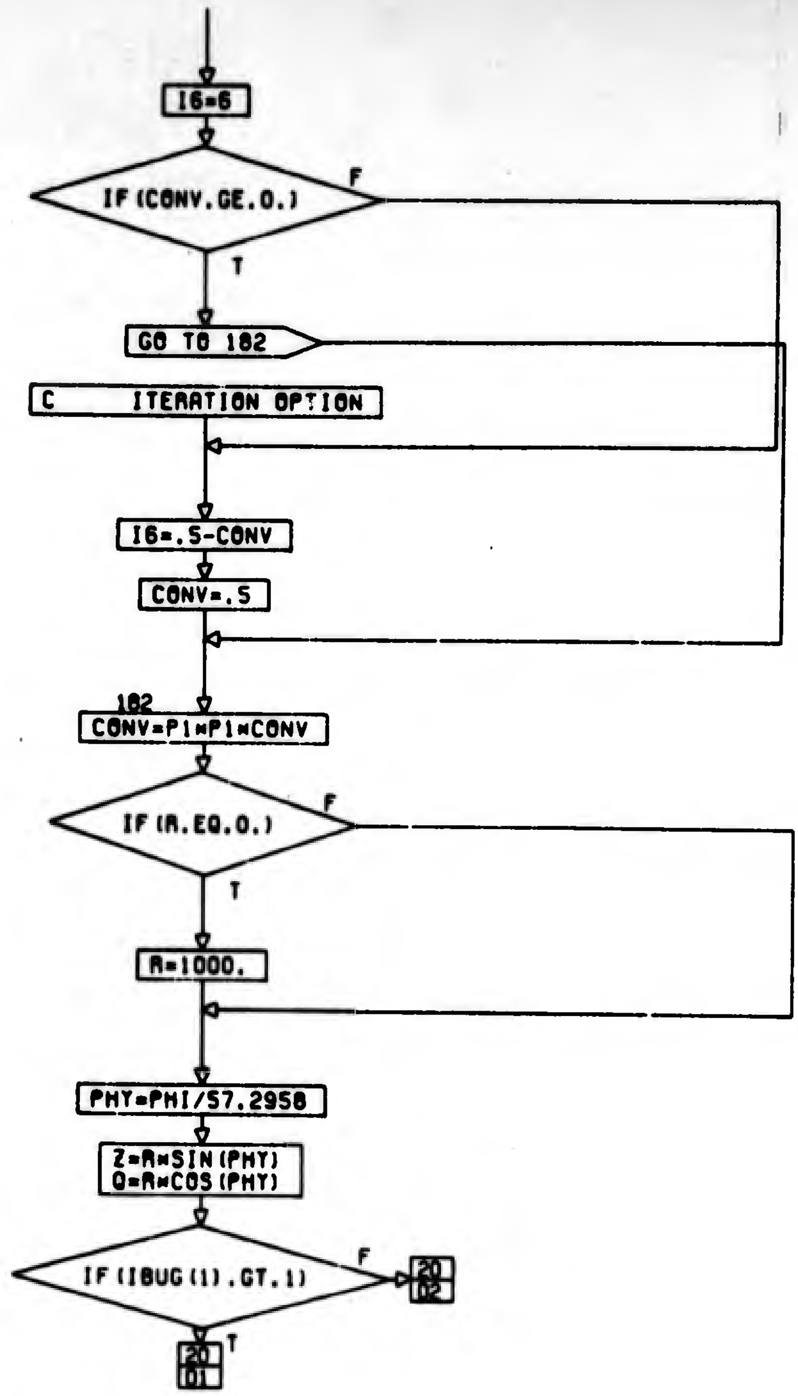


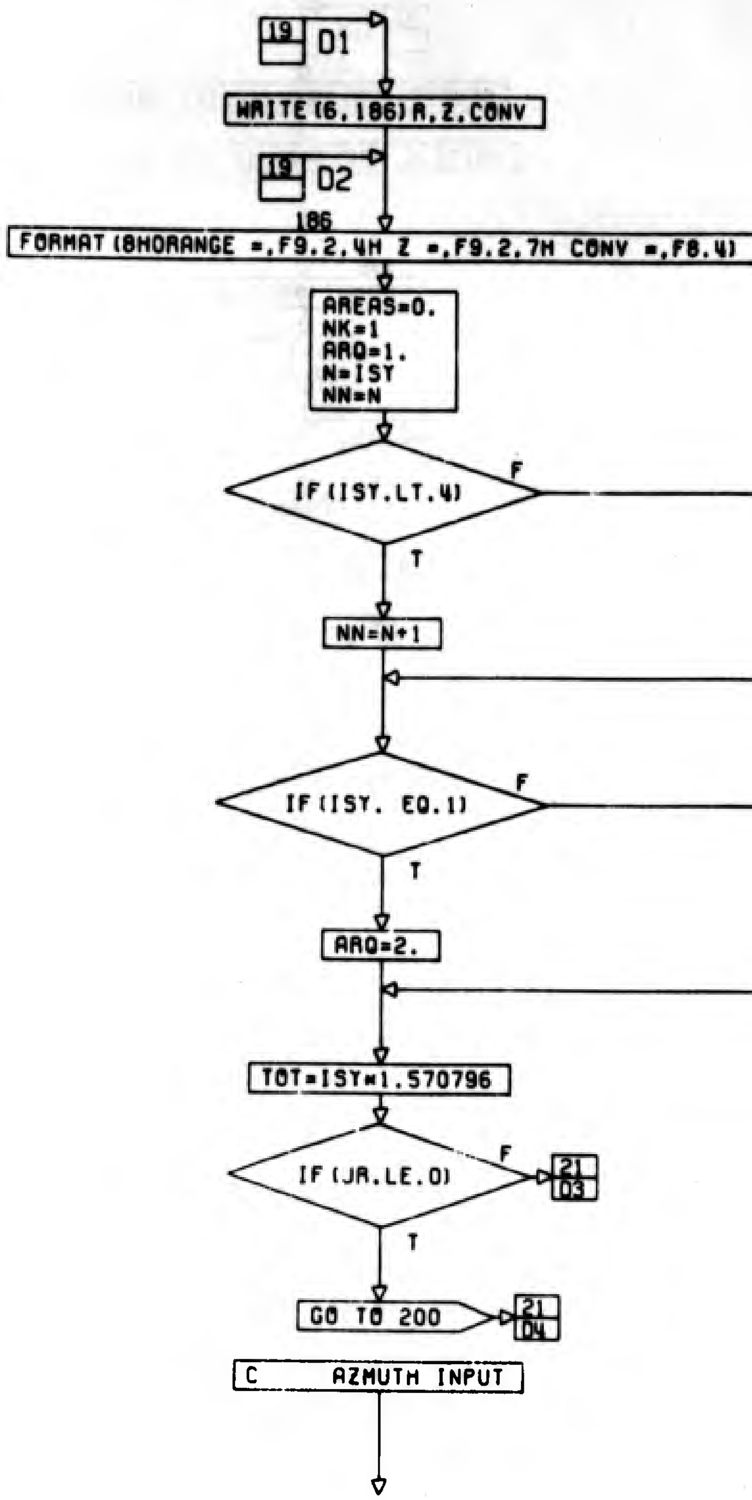
J=J+1
PVXYZ(J,1)=XX(M)
PVXYZ(J,2)=YY(M)
14
97
PVXYZ(J,3)=ZZ(M)
C STORE FACE
14
C4
12
98
CALL FACES(IFP,JPNT)
L1=L1+IST2-IST1+1
GO TO 27
5
A7
C ARCHITECTURAL PRINT OUT
5
C5
105
CONTINUE
LFCE(LL+1)=L+1
IF(JDID.NE.0)
F
T
GO TO 109
16
C6
C FIND RANGE OF X,Y,Z
DO 108 M=1,3
16
PMX(M)=PVXYZ(1,M)
PMN(M)=PVXYZ(1,M)
DO 108 I=2,J
16

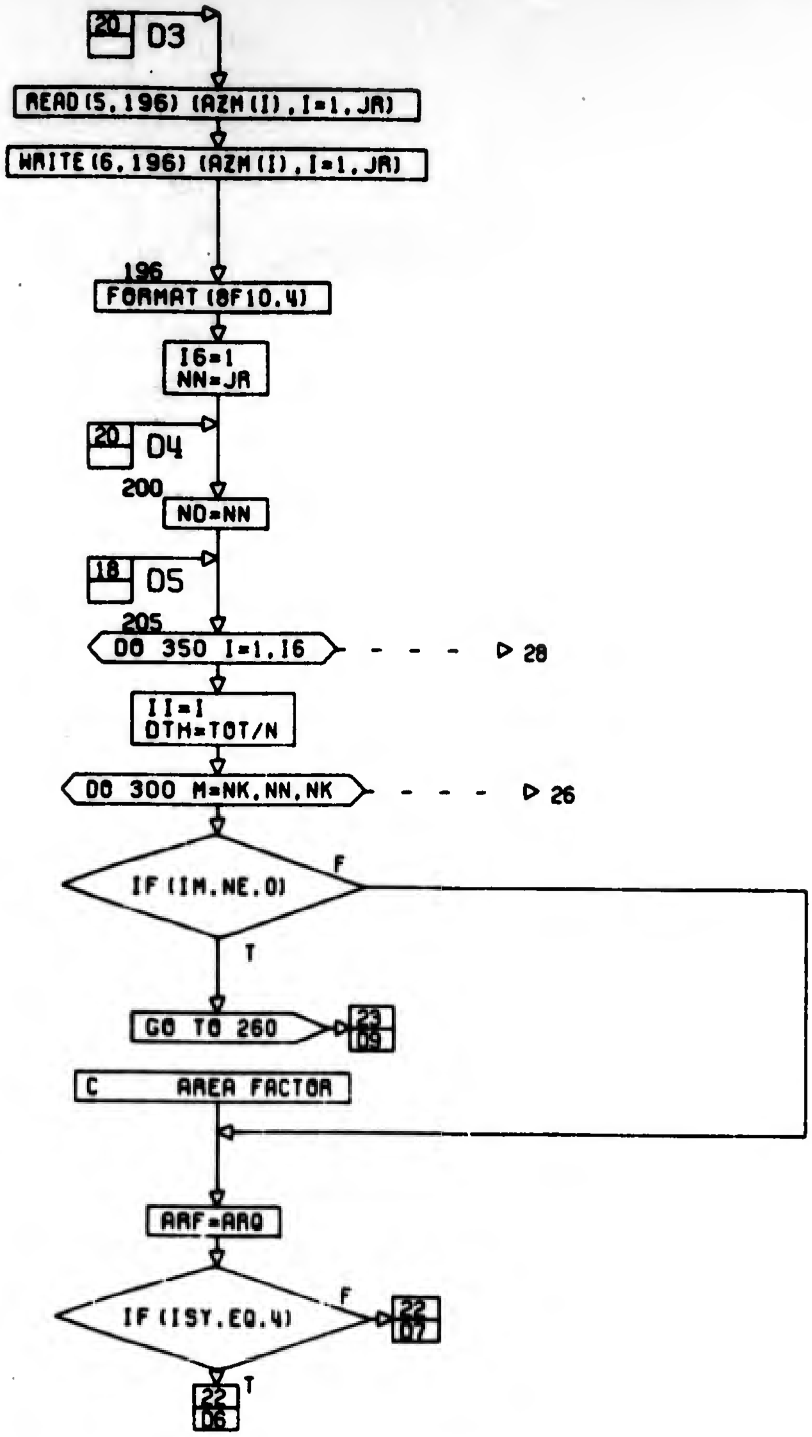

IF (PMX(M).LT.PVXYZ(I,M))
F
T
PMX(M)=PVXYZ(I,M)
IF (PMN(M).GT.PVXYZ(I,M))
F
T
PMN(M)=PVXYZ(I,M)
15,15
108
CONTINUE
15
C6
109
CONTINUE
IF (N.LT.2)
F
T
GO TO 150
17
C7
WRITE(6,110) (I,LFCE(I),LFCI(I),I=1,LL)
110
FORMAT(//7X25HFACE POINTER OBJECT/(3I10))

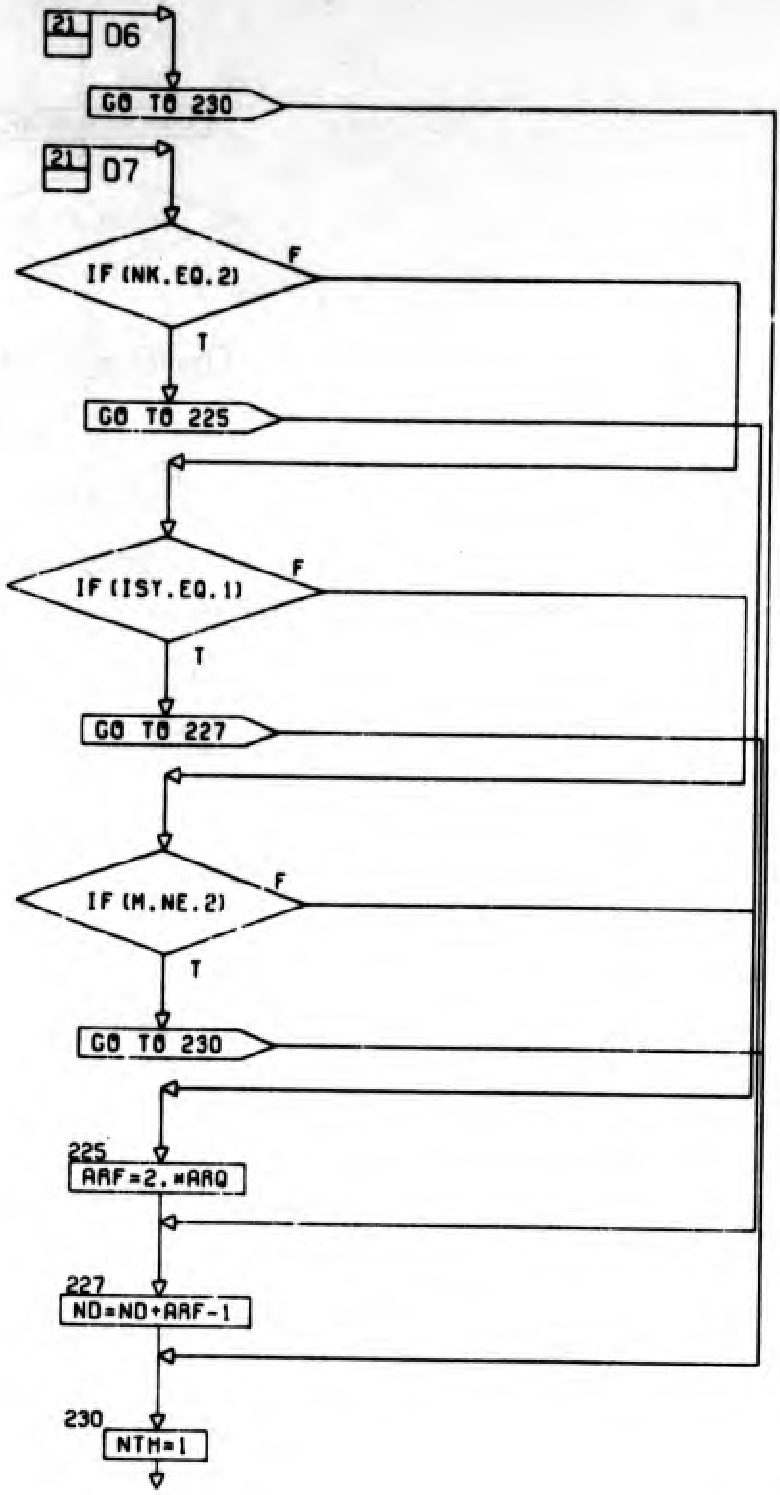

WRITE (6,120) (I, (PVXYZ (I,M), M=1,3), I=1, J)
120
FORMAT (//5H POINT, 6X1HX, 9X1HY, 9X1HZ/(I5, 3F10.2))
WRITE (6,125) (I, (JEDGE (I,M), M=1,5), I=1, NE)
125
FORMAT (//2X29HEDGE PT PT FACE FACE FLG/(6I5))
WRITE (6,130) (I, JFCE (I), JFCET (I), I=1, L)
130
FORMAT (//3X14HLOC EDGE POINT/(3I5))
WRITE (6,135) PMN, PMX
135
FORMAT (60H0********MINIMUM POINT******** ********MAXIMUM POINT****
****/6F10.2)
16
C7
C7
26
28
150
CONTINUE
READ (5,134) P1, P2, PHI, CONV, JR, IM, IREA, R, SY, IPLOT, IBUG, P3, INOBJ
WRITE (6,134) P1, P2, PHI, CONV, JR, IM, IREA, R, SY, IPLOT, IBUG, P3, INOBJ
134
FORMAT (2F5.2, 2F10.3, I2, 2I1, F6.2, A1, 5I1, 1X, F6.2, 2X, 15I1)
IF (P1.EQ.0.)
F
16
C9
T
18
C8

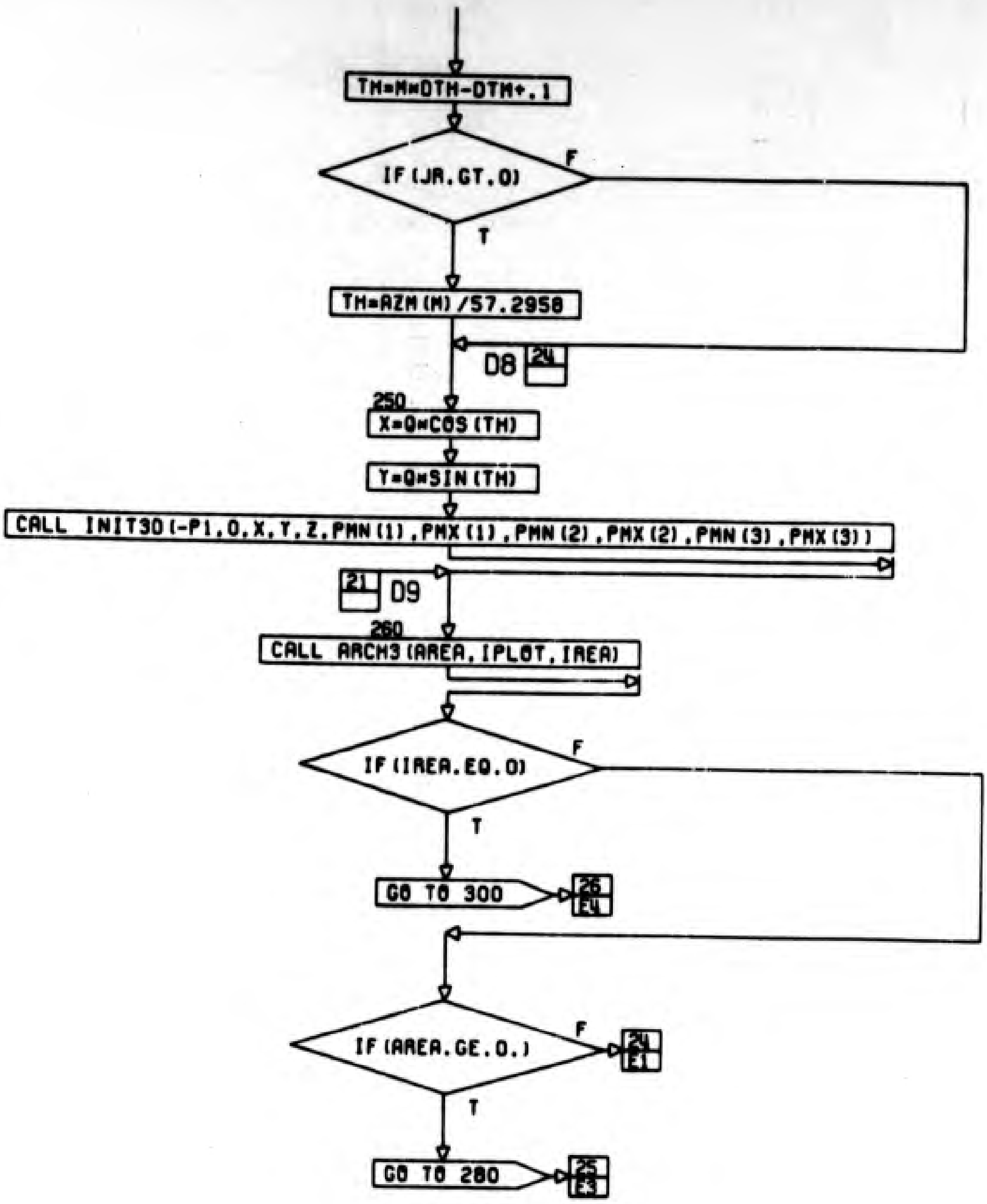

17
C8
GO TO 5
1
A1
17
C9
IF (P3.NE.0.)
F
T
CALL PLOTCC (P3,0.,-3)
IF (IM.NE.0)
F
T
GO TO 205
21
D5
C CHECK SYMMETRY
DO 160 ISY=1,2
IF (SY.EQ.SYM(ISY))
F
T
GO TO 170
160
CONTINUE
ISY=4
170
CONTINUE

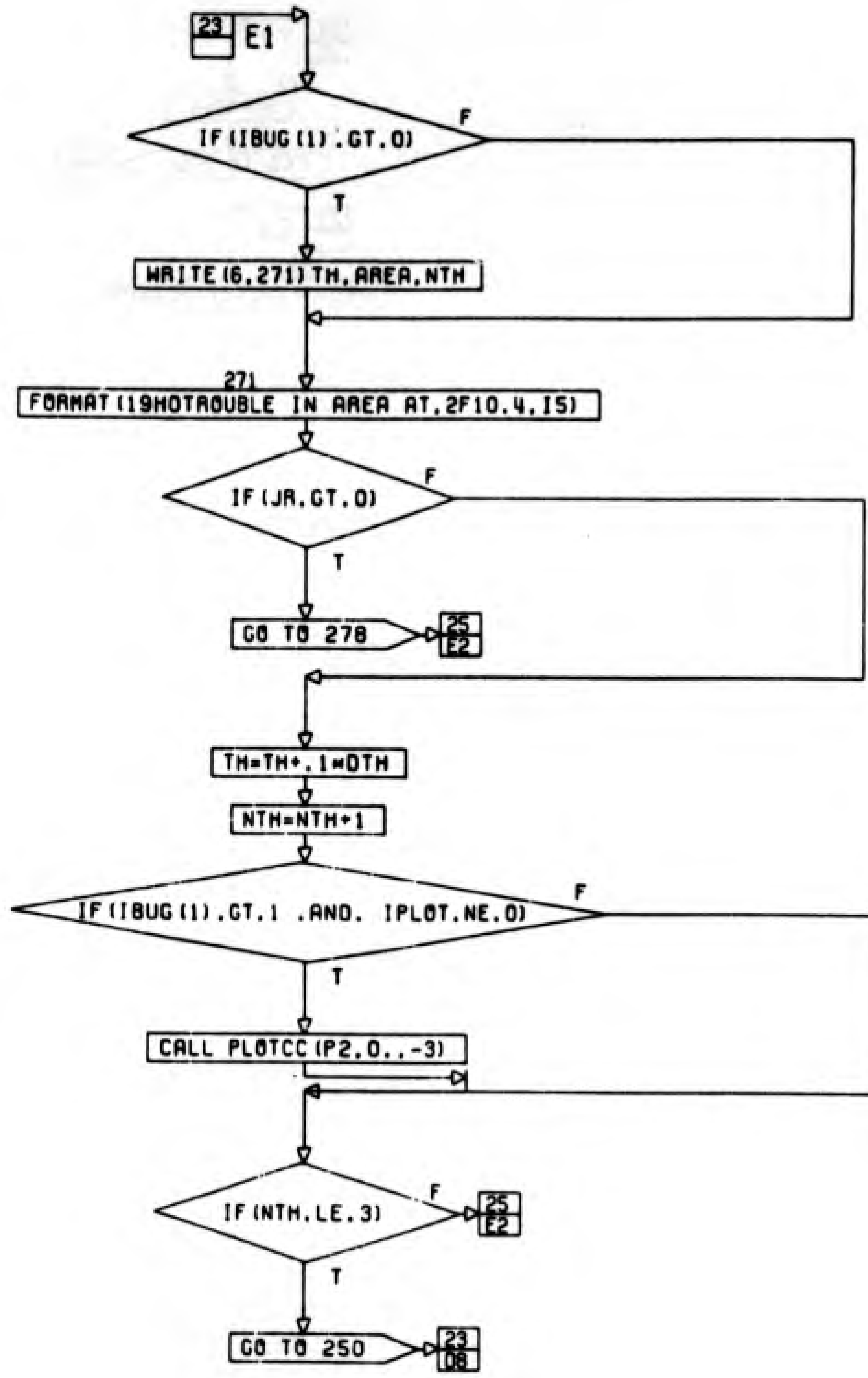

I6=6
IF (CONV.GE.O.)
F
T
GO TO 182
C ITERATION OPTION
I6=.5-CONV
CONV=.5
182
CONV=PI*PI*CONV
IF (R.EQ.O.)
F
T
R=1000.
PHY=PHI/57.2958
Z=R*SIN(PHY)
Q=R*COS(PHY)
IF (IBUG(1).GT.1)
F
20
02
T
20
01

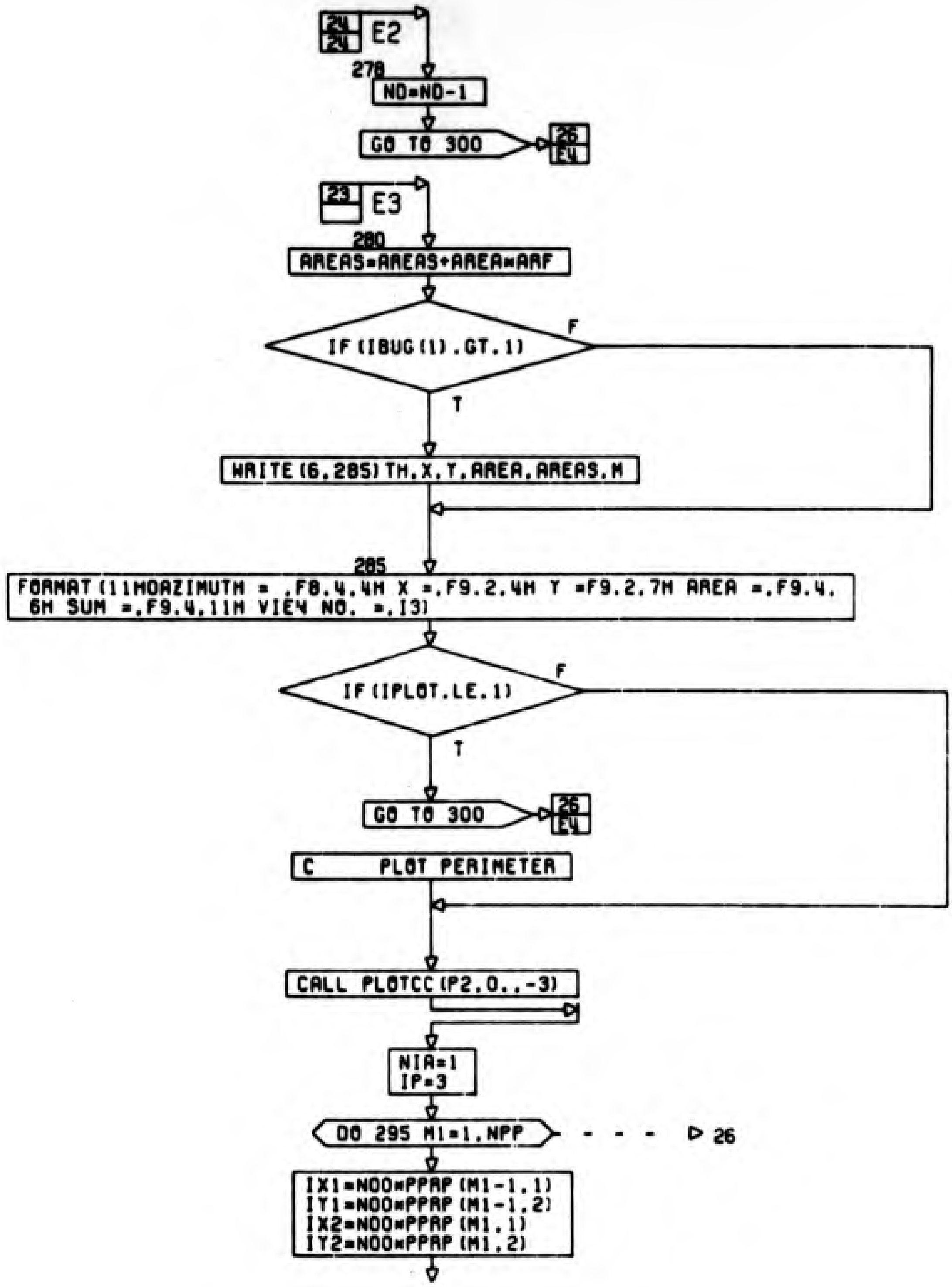

19 D1
WRITE(6,186)R,Z,CONV
19 D2
186
FORMAT(8HORANGE =,F9.2,4H Z =,F9.2,7H CONV =,F8.4)
AREAS=0.
NK=1
ARQ=1.
N=ISY
NN=N
IF(ISY.LT.4)
F
T
NN=N+1
IF(ISY. EQ.1)
F
T
ARQ=2.
TOT=ISY*1.570796
IF(JR.LE.0)
F
21 D3
T
GO TO 200
21 D4
C AZMUTH INPUT


20
D3
READ(5,196) (AZM(I),I=1,JR)
WRITE(6,196) (AZM(I),I=1,JR)
196
FORMAT(8F10.4)
I6=1
NN=JR
20
D4
200
ND=NN
18
D5
205
DO 350 I=1,I6
28
II=I
DTM=TOT/N
DO 300 M=NK,NN,NK
26
IF(IM.NE.0)
F
T
GO TO 260
23
D9
C AREA FACTOR
ARF=ARO
IF(ISY.EQ.4)
F
22
D7
T
22
D6

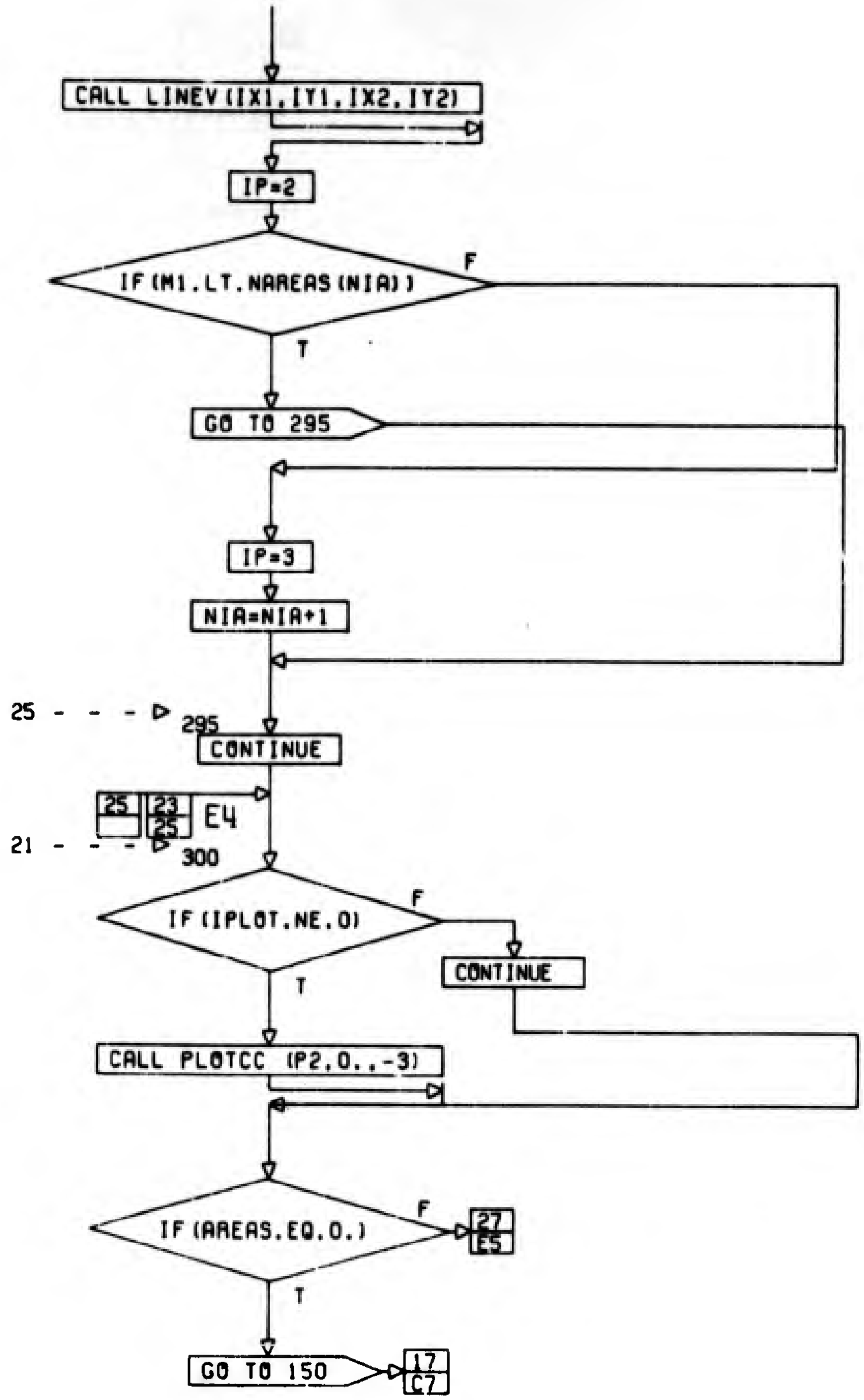

21
D6
GO TO 230
21
D7
IF (NK.EQ.2)
F
T
GO TO 225
IF (ISY.EQ.1)
F
T
GO TO 227
IF (M.NE.2)
F
T
GO TO 230
225
ARF=2.*ARQ
227
ND=ND+ARF-1
230
NTH=1

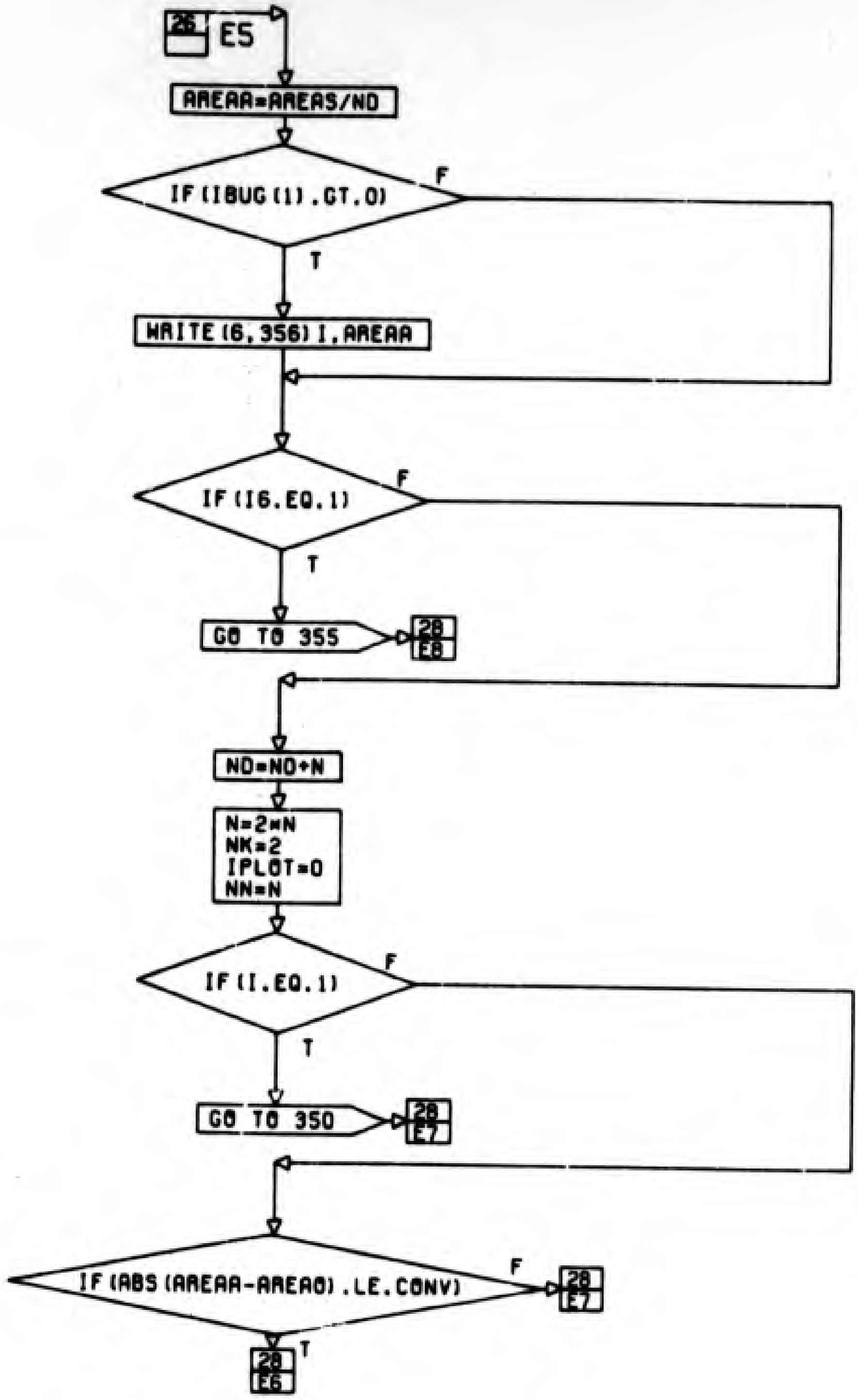

TM=M*DTM-DTM+.1
IF (JR.GT.0)
F
T
TM=AZM(M)/57.2958
D8
24
250
X=Q*COS(TM)
Y=Q*SIN(TM)
CALL INIT3D(-P1,0,X,Y,Z,PMN(1),PMX(1),PMN(2),PMX(2),PMN(3),PMX(3))
21
D9
260
CALL ARCH3(AREA,IPLOT,IAREA)
IF (IAREA.EQ.0)
F
T
GO TO 300
26
E4
IF (AREA.GE.0.)
F
24
E1
T
GO TO 280
25
E3


23
E1
IF (IBUG (1) .GT. 0)
F
T
WRITE (6, 271) TH, AREA, NTH
271
FORMAT (19HOTROUBLE IN AREA AT, 2F10.4, I5)
IF (JR.GT.0)
GO TO 278
25
E2
TH=TH+.1*DTH
NTH=NTH+1
IF (IBUG (1).GT.1 .AND. IPLOT.NE.0)
CALL PLOTCC (P2.0.,-3)
IF (NTH.LE.3)
GO TO 250
23
D8

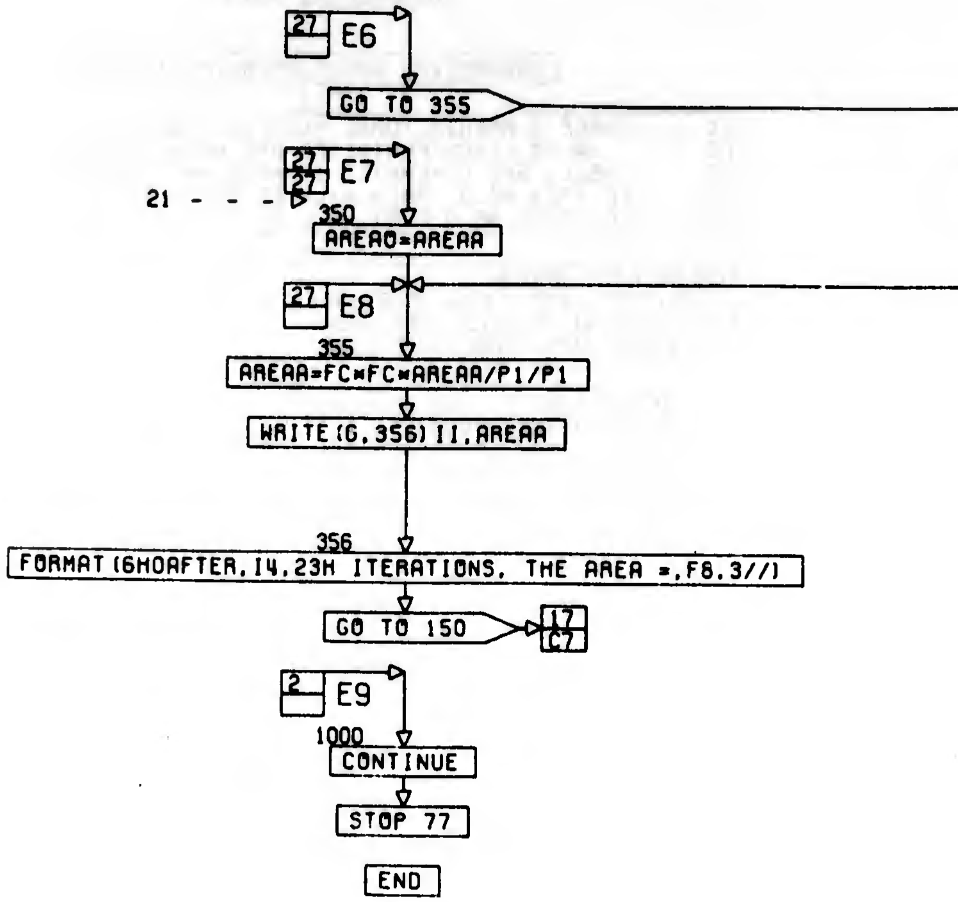


24
24
E2
278
ND=ND-1
GO TO 300
26
E4
23
E3
280
AREAS=AREAS+AREA*ARF
IF (IBUG(1).GT.1)
F
T
WRITE(6,285) TH,X,Y,AREA,AREAS,M
285
FORMAT(11H0AZIMUTH = ,F8.4,4H X =,F9.2,4H Y =F9.2,7H AREA =,F9.4,
6H SUM =,F9.4,11H VIEW NO. =,I3)
IF (IPLOT.LE.1)
F
T
GO TO 300
26
E4
C PLOT PERIMETER
CALL PLOTCC(P2,0.,-3)
NIA=1
IP=3
DO 295 M1=1,NPP
26
IX1=NOO*PPRP(M1-1,1)
IY1=NOO*PPRP(M1-1,2)
IX2=NOO*PPRP(M1,1)
IY2=NOO*PPRP(M1,2)

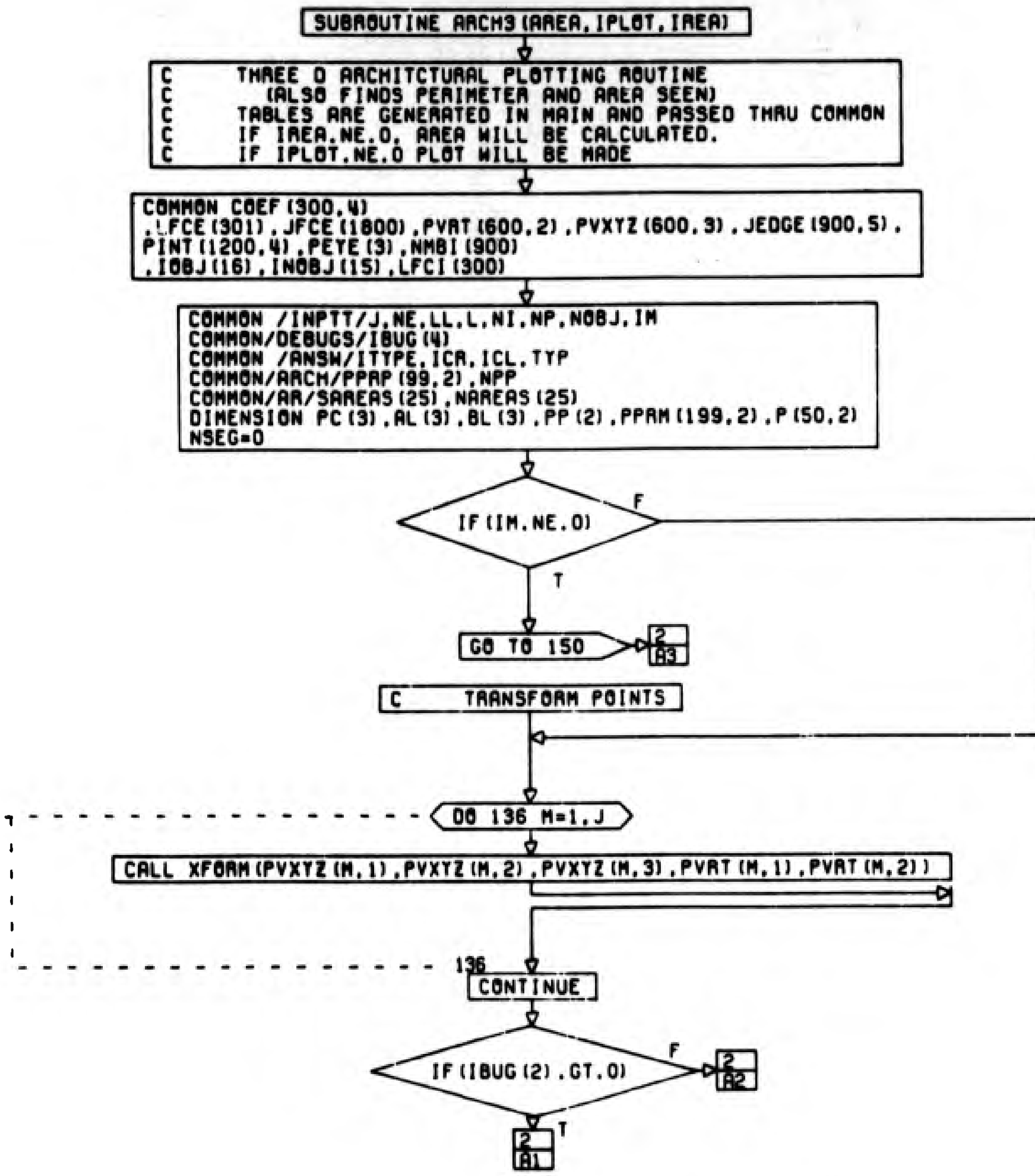

CALL LINEV(IX1,IY1,IX2,IY2)
IP=2
IF(M1.LT.NAREAS(NIA))
F
T
GO TO 295
IP=3
NIA=NIA+1
25
295
CONTINUE
25
23
25
E4
21
300
IF(IPLOT.NE.0)
F
T
CONTINUE
CALL PLOTCC (P2,0.,-3)
IF(AREAS.EQ.0.)
F
27
E5
T
GO TO 150
17
C7

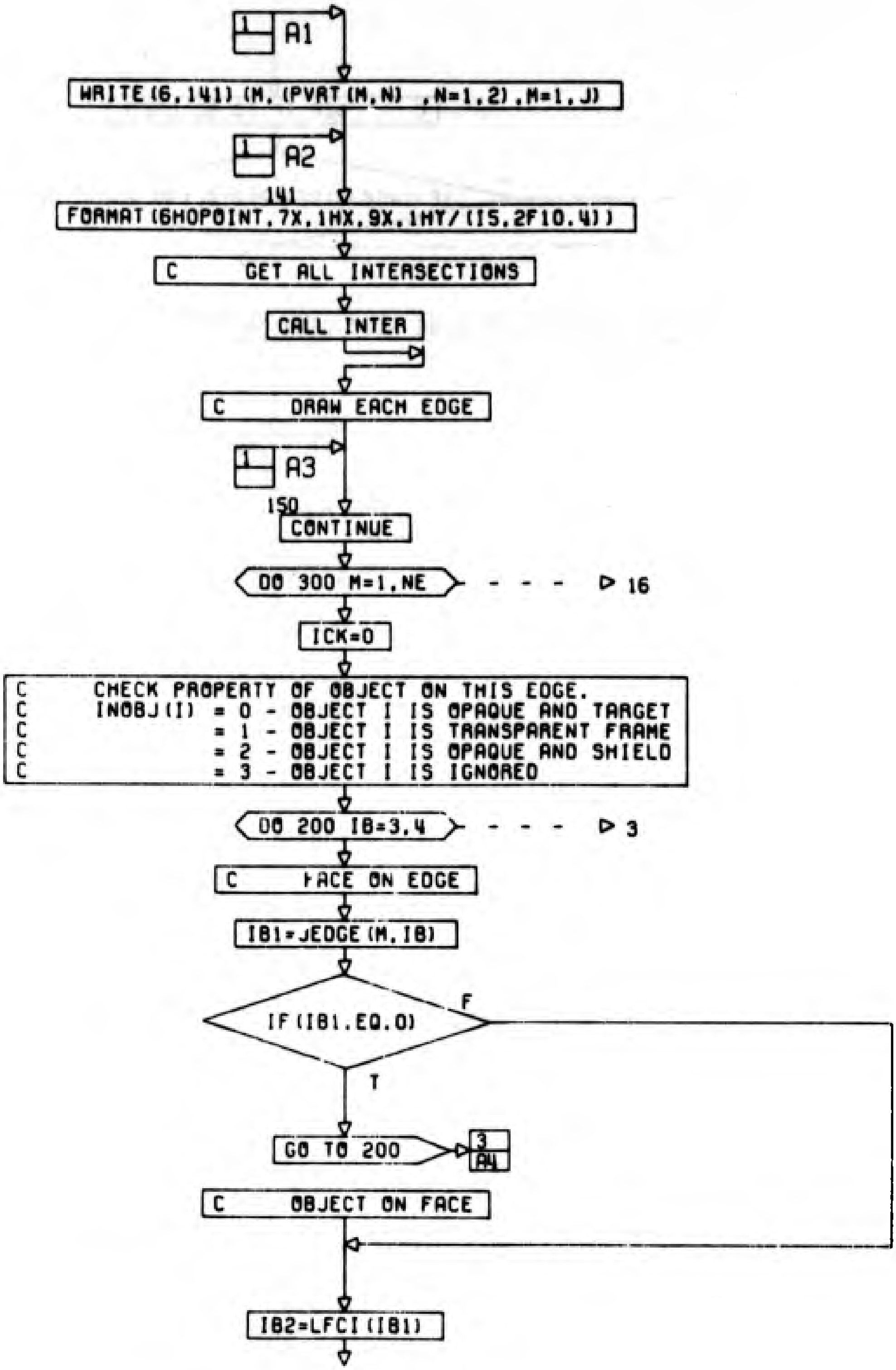


26
E5
AREAA=AREAS/ND
IF(IBUG(1).GT.0)
F
T
WRITE(6,356)I,AREAA
IF(I6.EQ.1)
F
T
GO TO 355
28
E8
ND=ND+N
N=2*N
NK=2
IPLOT=0
NN=N
IF(I.EQ.1)
F
T
GO TO 350
28
E7
IF(ABS(AREAA-AREAO).LE.CONV)
F
28
E7
T
28
E6

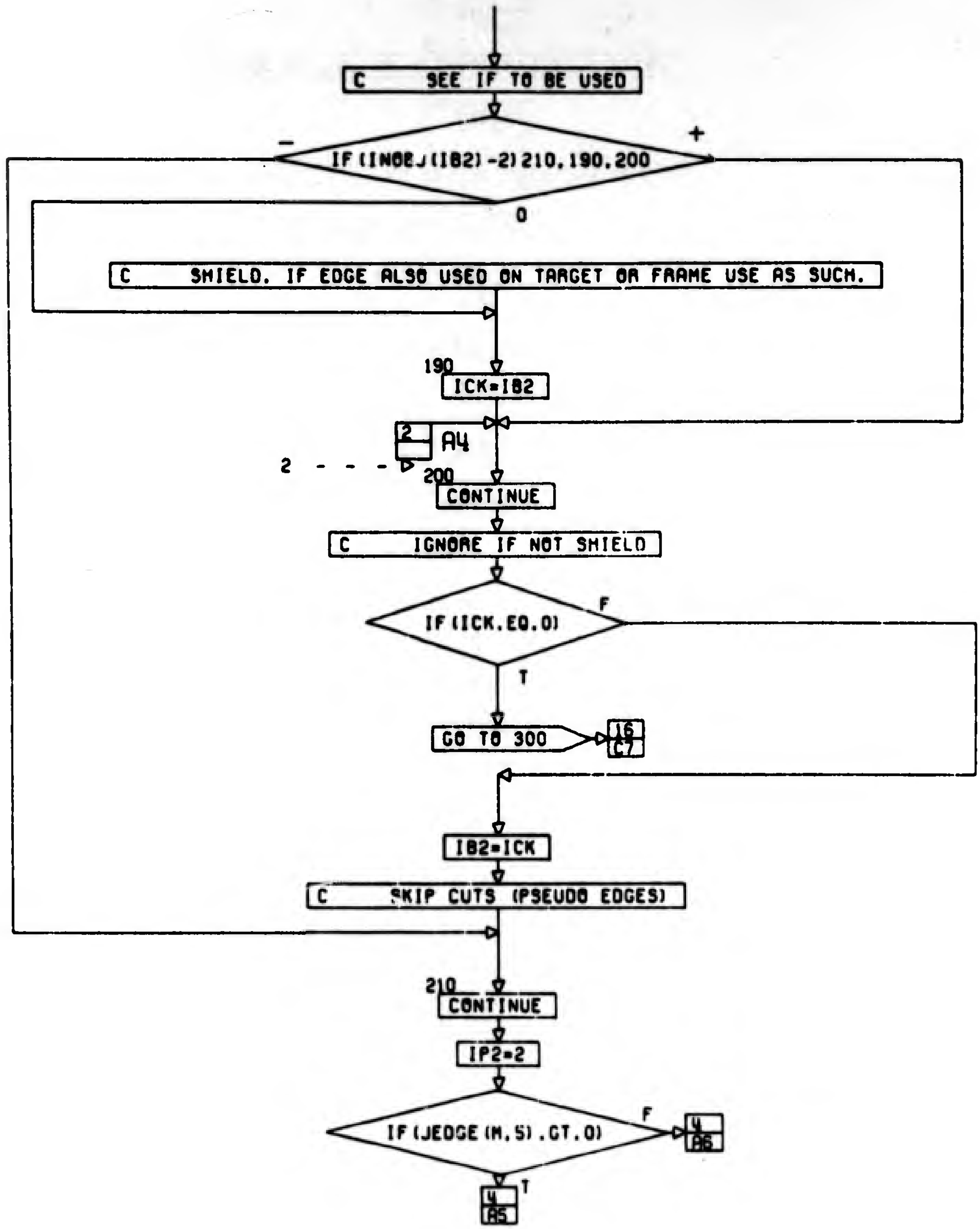

27 E6
GO TO 355
27 27 E7
21
350
AREAO=AREAA
27 E8
355
AREAA=FC*FC*AREAA/P1/P1
WRITE(6,356) II,AREAA
356
FORMAT(6HOAFTER,I4,23H ITERATIONS, THE AREA =,F8.3//)
GO TO 150
17 C7
2 E9
1000
CONTINUE
STOP 77
END

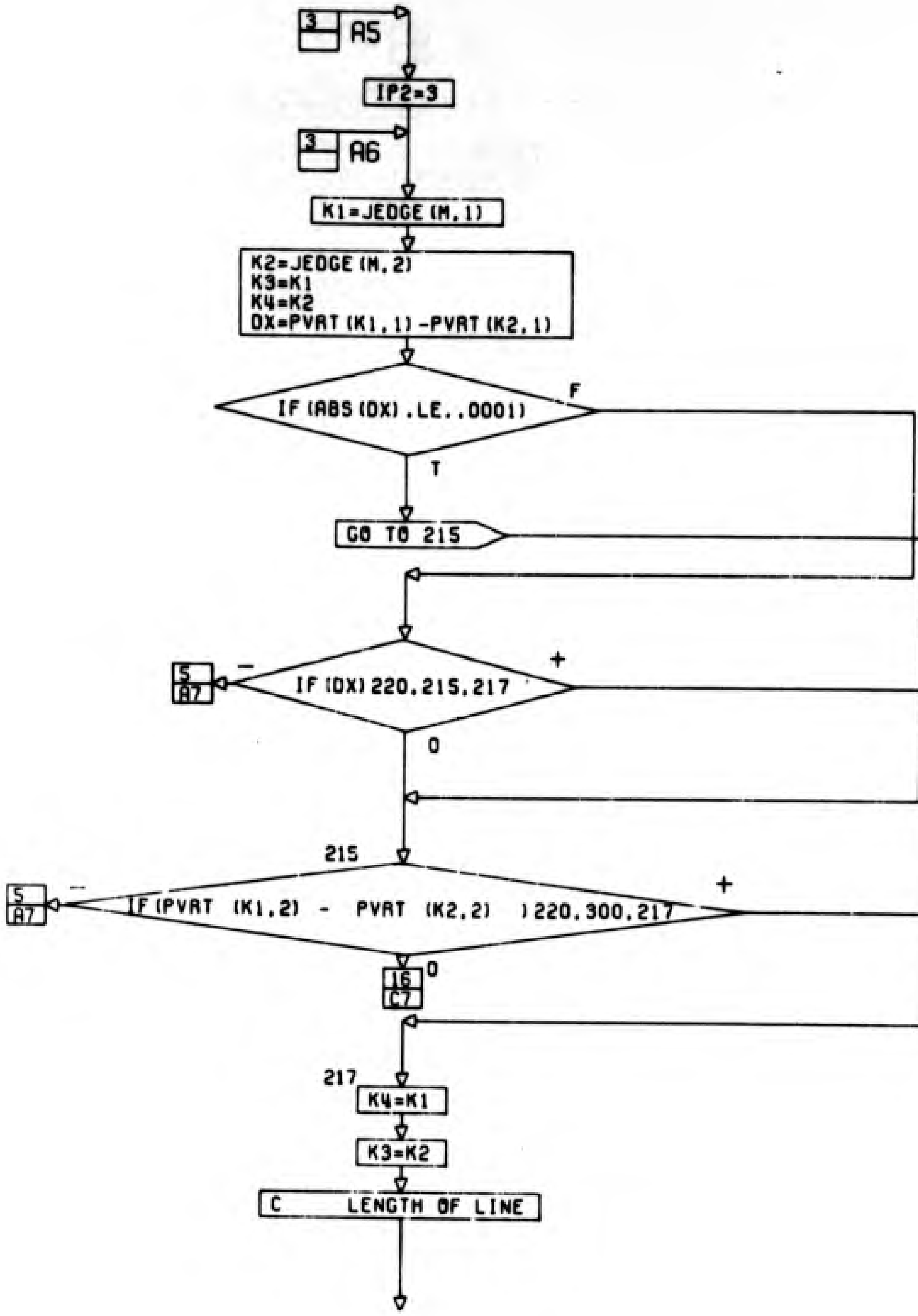
SUBROUTINE ARCH3

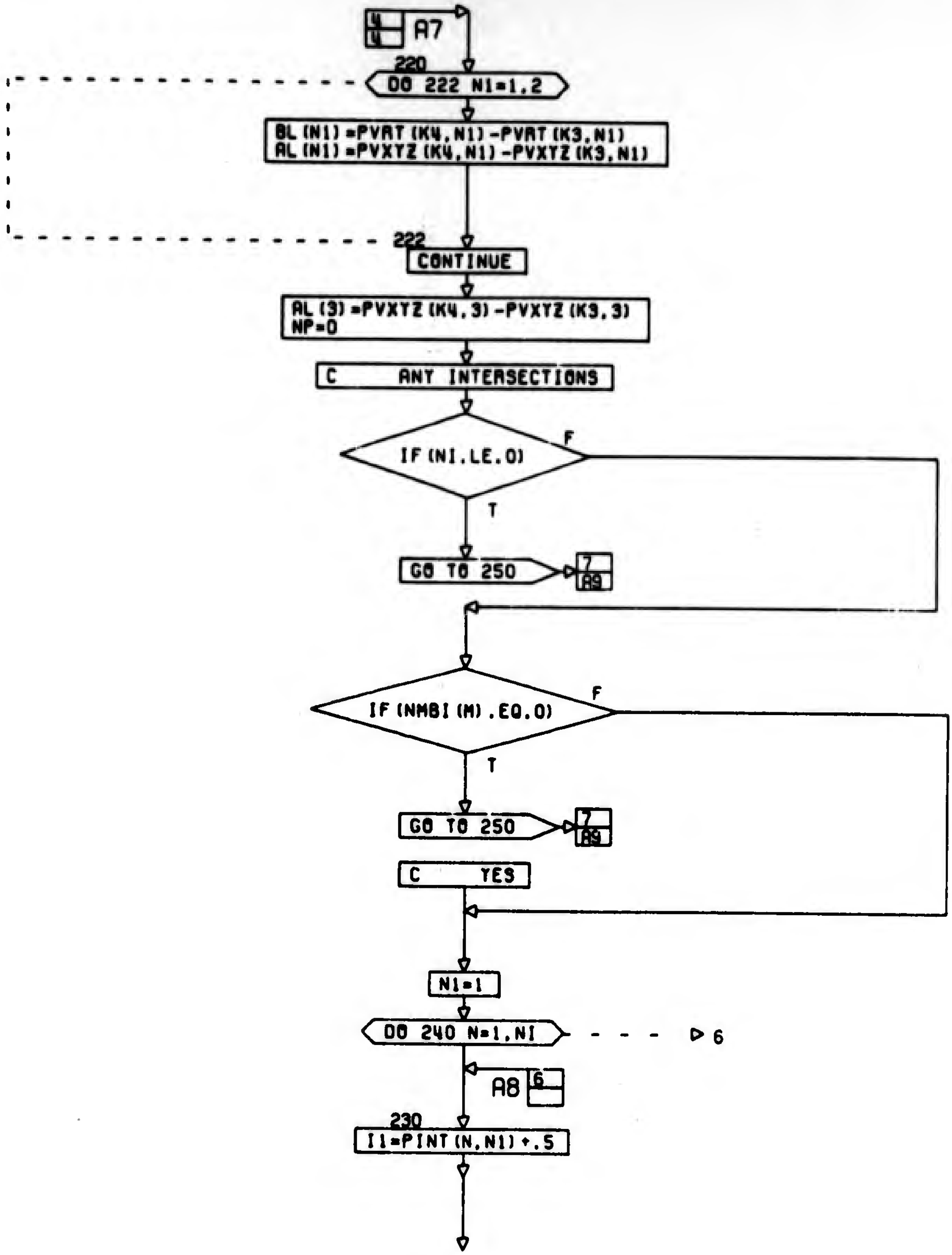




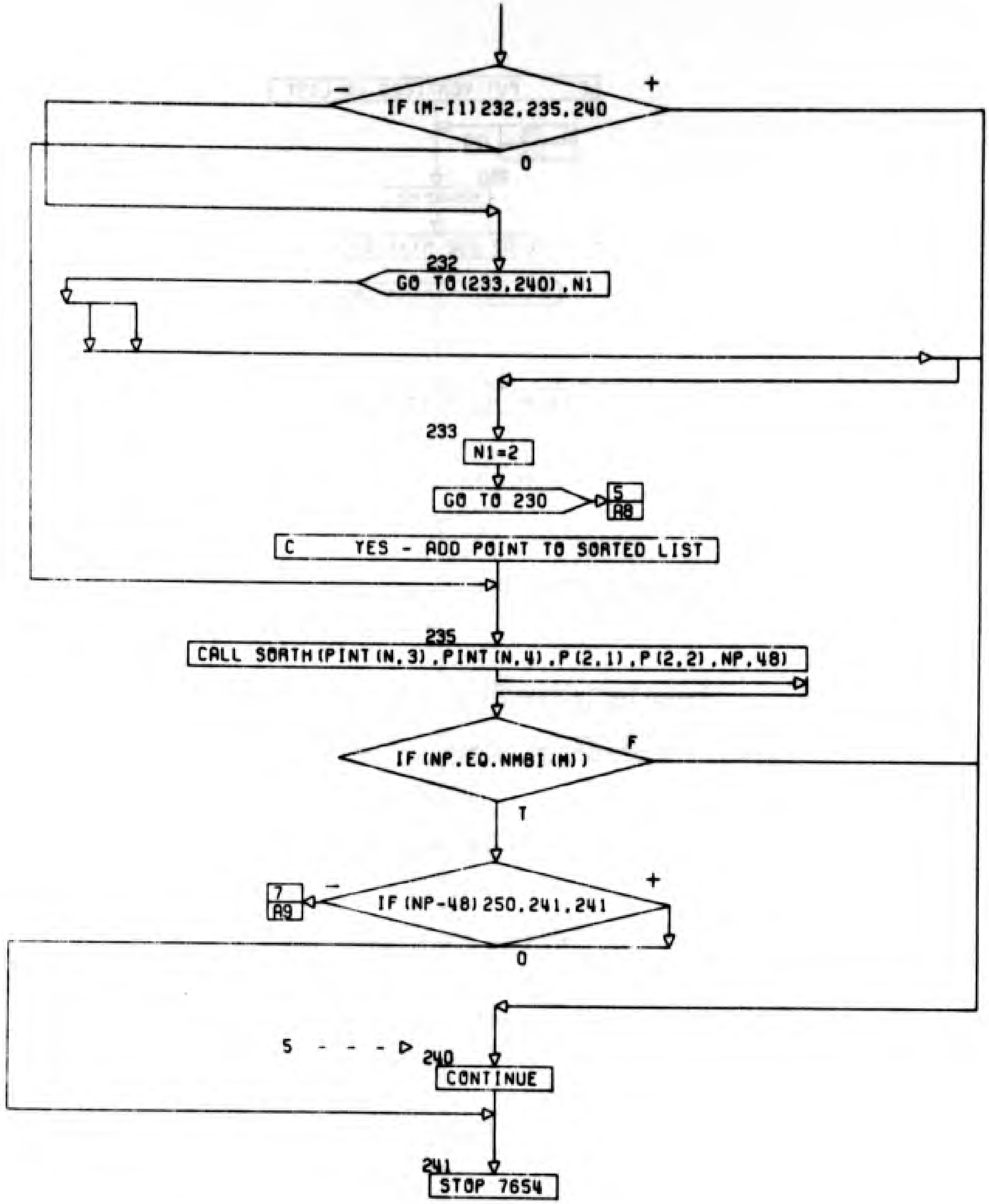
1 A1
WRITE (6,141) (M, (PVRT (M,N) ,N=1,2), M=1, J)
1 A2
141
FORMAT (6HOPOINT, 7X, 1HX, 9X, 1HY/ (I5, 2F10.4))
C GET ALL INTERSECTIONS
CALL INTER
C DRAW EACH EDGE
1 A3
150
CONTINUE
DO 300 M=1, NE
16
ICK=0
C CHECK PROPERTY OF OBJECT ON THIS EDGE.
C INOBJ(I) = 0 - OBJECT I IS OPAQUE AND TARGET
C = 1 - OBJECT I IS TRANSPARENT FRAME
C = 2 - OBJECT I IS OPAQUE AND SHIELD
C = 3 - OBJECT I IS IGNORED
DO 200 IB=3,4
3
C FACE ON EDGE
IB1=JEDGE (M, IB)
IF (IB1.EQ.0)
F
T
GO TO 200
3
A4
C OBJECT ON FACE
IB2=LFCI (IB1)


C SEE IF TO BE USED
IF (INOBJ(IB2)-2)210,190,200
-
+
0
C SHIELD. IF EDGE ALSO USED ON TARGET OR FRAME USE AS SUCH.
190
ICK=IB2
2
A4
2
200
CONTINUE
C IGNORE IF NOT SHIELD
IF(ICK.EQ.0)
F
T
GO TO 300
16
C7
IB2=ICK
C SKIP CUTS (PSEUDO EDGES)
210
CONTINUE
IP2=2
IF(JEDGE(M,5).GT.0)
F
4
A6
T
4
A5

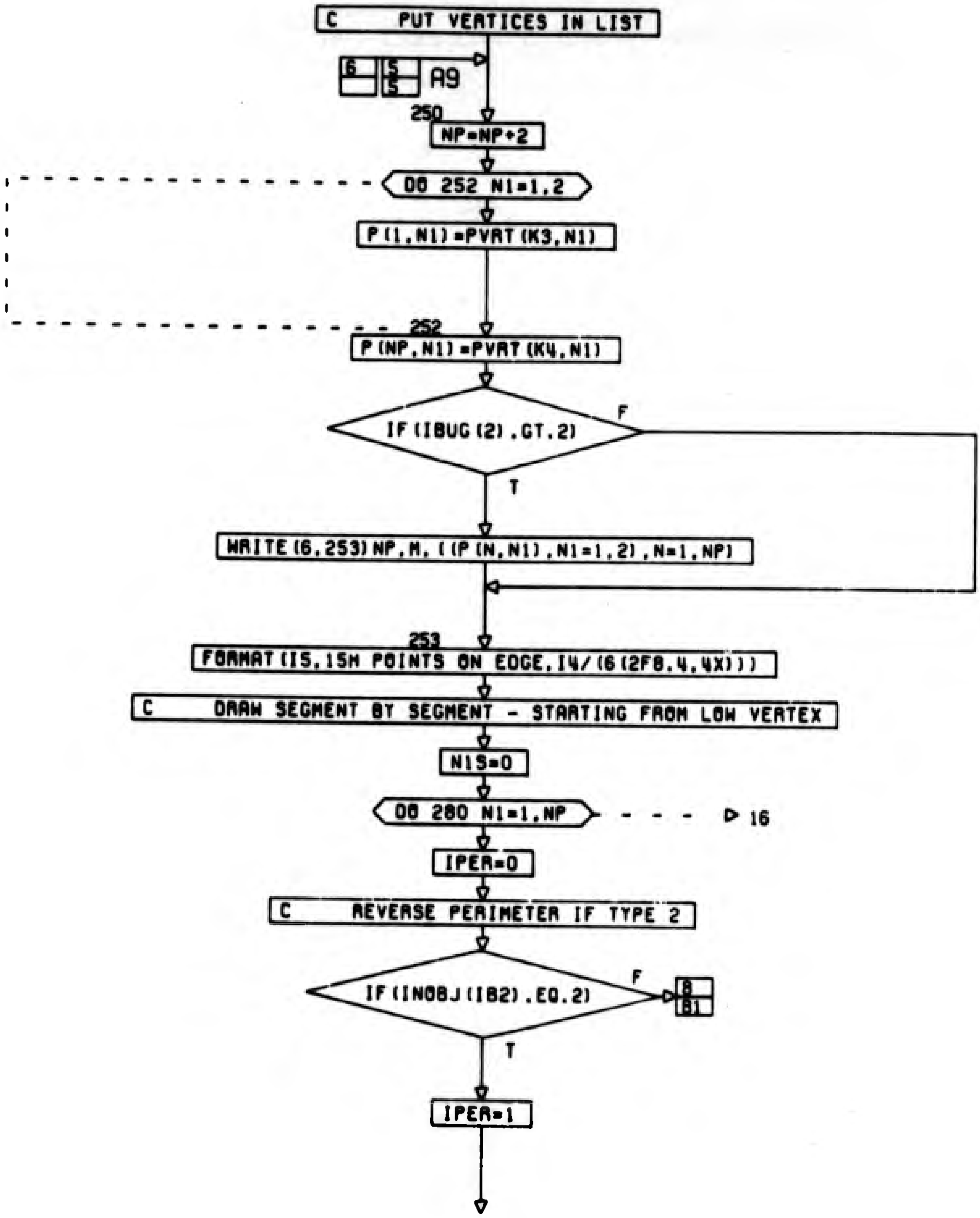

3
A5
IP2=3
3
A6
K1=JEDGE(M,1)
K2=JEDGE(M,2)
K3=K1
K4=K2
DX=PVAT(K1,1)-PVAT(K2,1)
IF(ABS(DX).LE..0001)
F
T
GO TO 215
IF(DX)220,215,217
−
+
0
5
A7
215
IF(PVAT (K1,2) - PVAT (K2,2))220,300,217
−
+
0
5
A7
16
C7
217
K4=K1
K3=K2
C LENGTH OF LINE

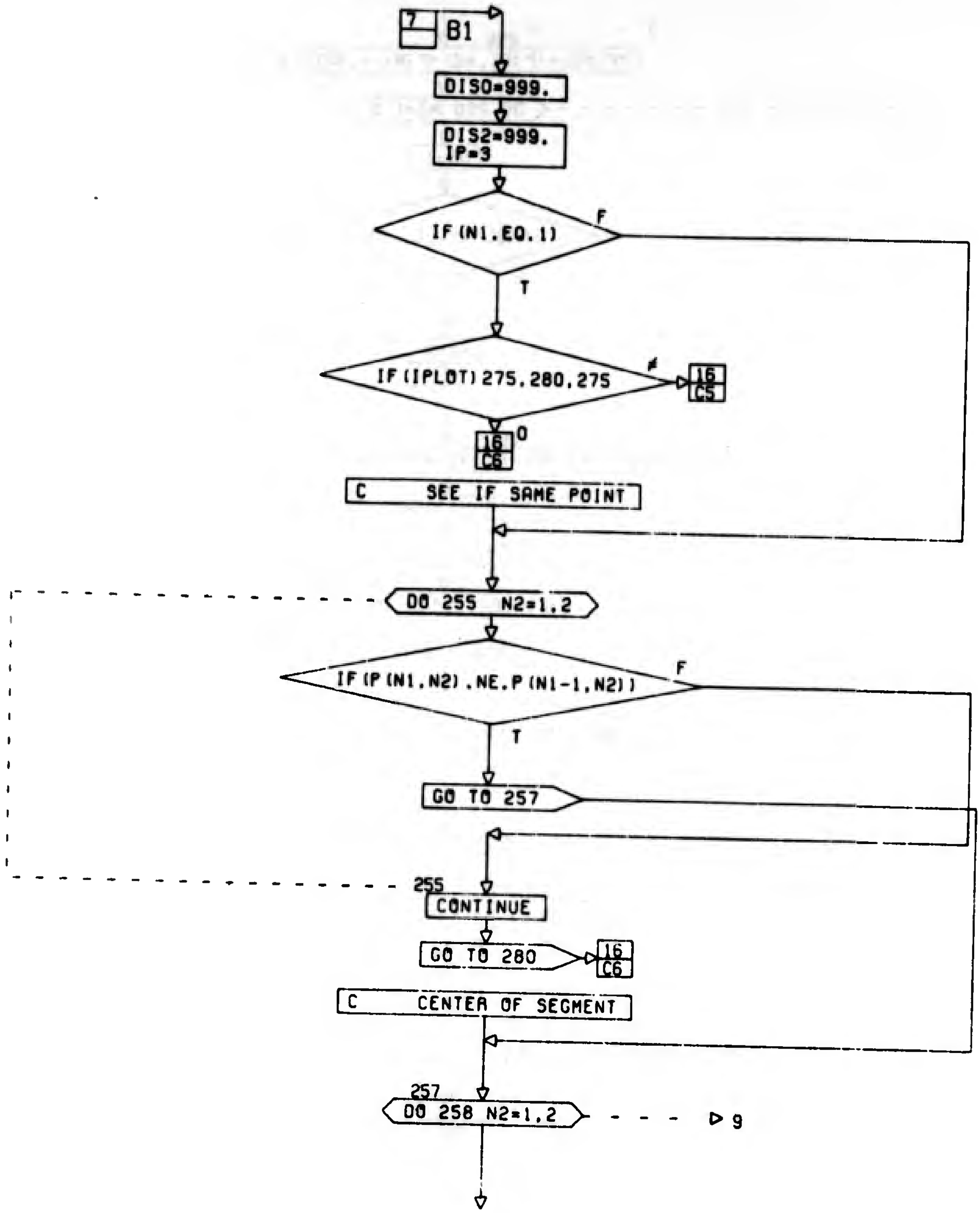


A7
220
DO 222 N1=1,2
BL(N1)=PVRT(K4,N1)-PVRT(K3,N1)
AL(N1)=PVXYZ(K4,N1)-PVXYZ(K3,N1)
222
CONTINUE
AL(3)=PVXYZ(K4,3)-PVXYZ(K3,3)
NP=0
C ANT INTERSECTIONS
IF(NI.LE.0)
F
T
GO TO 250
7
A9
IF(NMBI(M).EQ.0)
F
T
GO TO 250
7
A9
C YES
N1=1
DO 240 N=1,NI
6
A8
6
230
I1=PINT(N,N1)+.5

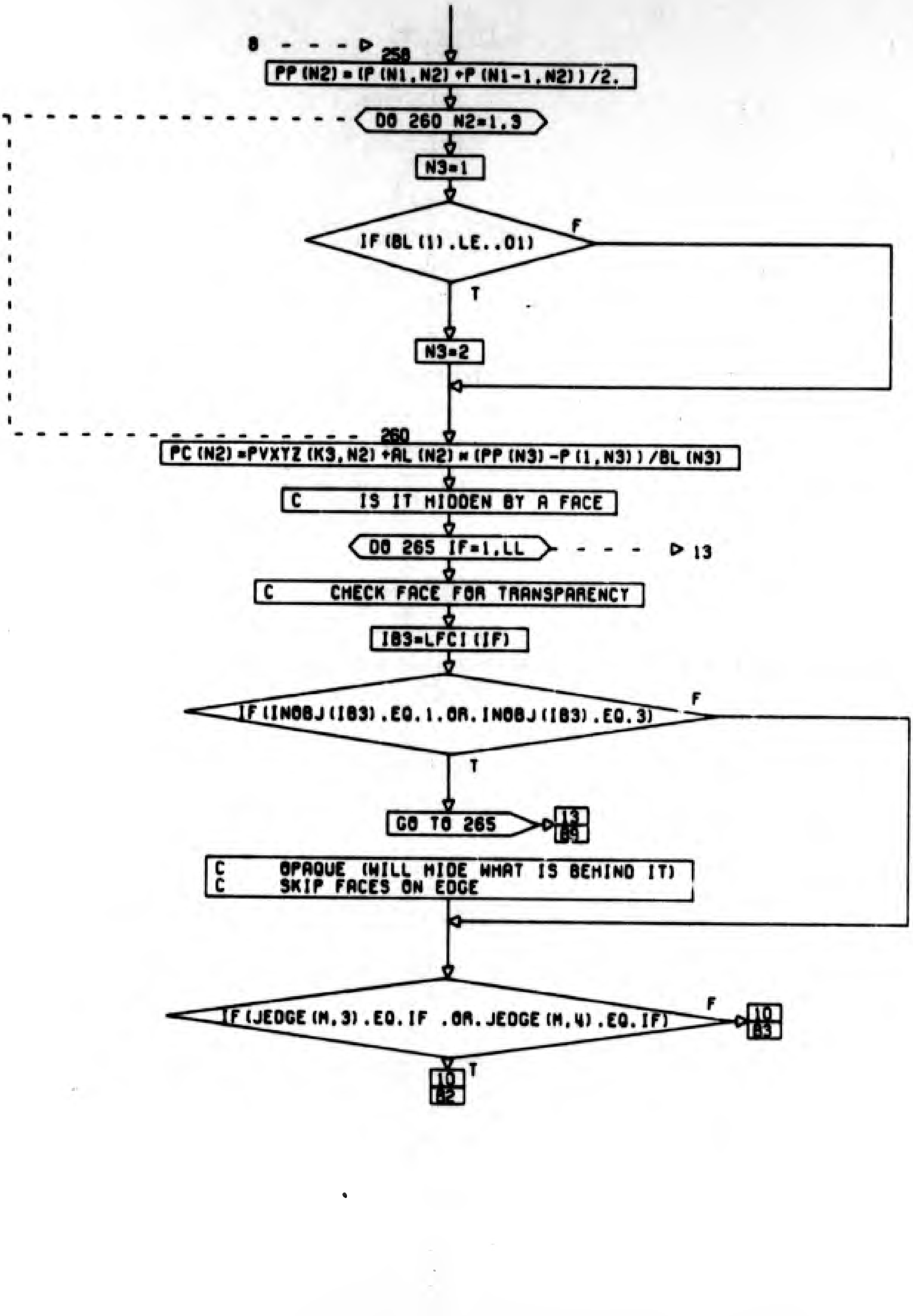


IF (M-I1) 232,235,240
-
+
0
232
GO TO (233,240),N1
233
N1=2
GO TO 230
5
A8
C YES - ADD POINT TO SORTED LIST
235
CALL SORTH(PINT(N,3),PINT(N,4),P(2,1),P(2,2),NP,48)
IF (NP.EQ.NMBI(M))
F
T
IF (NP-48) 250,241,241
7
A9
-
+
0
5
240
CONTINUE
241
STOP 7654

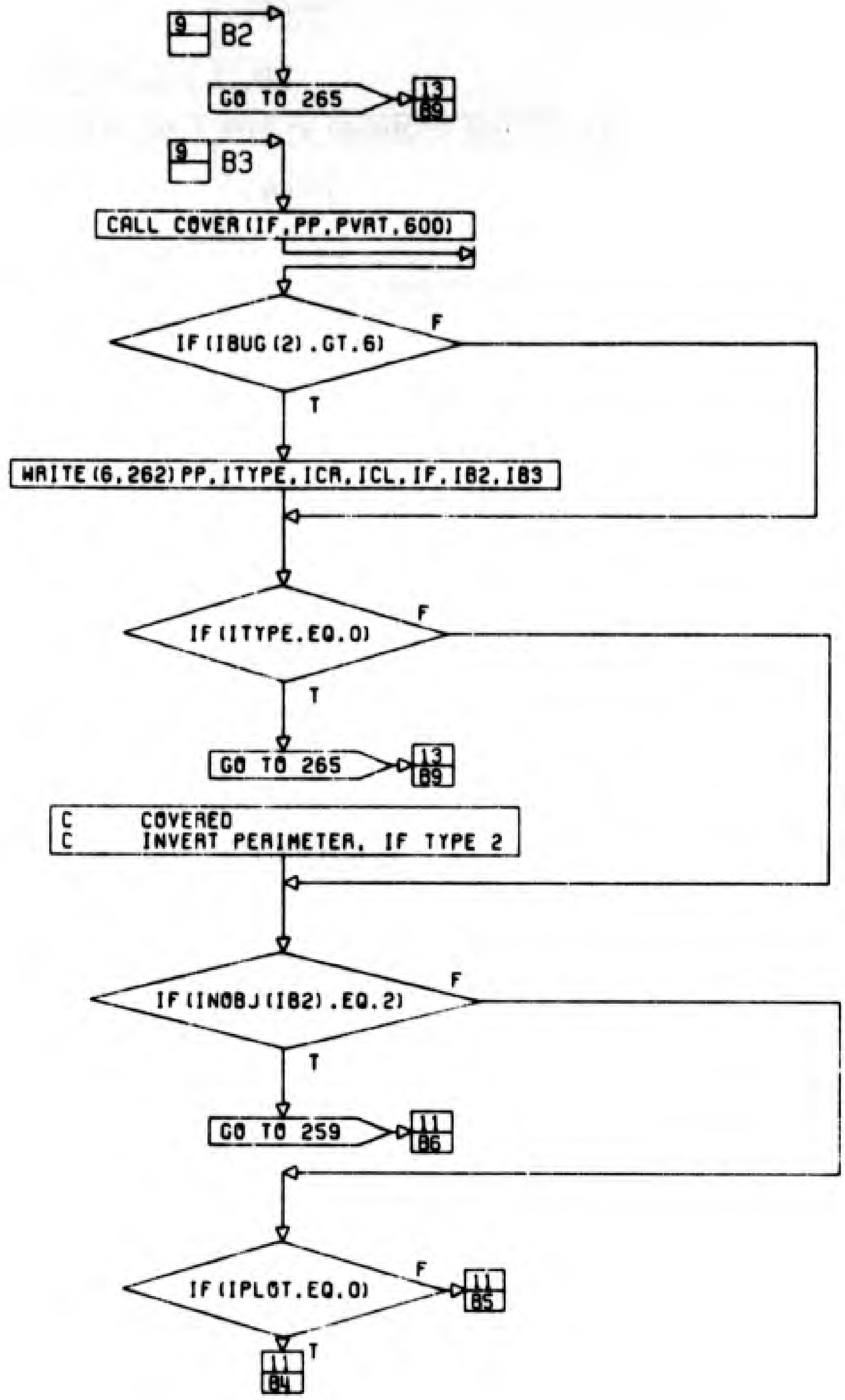

C PUT VERTICES IN LIST
6
5
5
A9
250
NP=NP+2
DO 252 N1=1,2
P(1,N1)=PVRT(K3,N1)
252
P(NP,N1)=PVRT(K4,N1)
IF(IBUG(2).GT.2)
F
T
WRITE(6,253)NP,M,((P(N,N1),N1=1,2),N=1,NP)
253
FORMAT(I5,15H POINTS ON EDGE,I4/(6(2F8.4,4X)))
C DRAW SEGMENT BY SEGMENT - STARTING FROM LOW VERTEX
N1S=0
DO 280 N1=1,NP
16
IPER=0
C REVERSE PERIMETER IF TYPE 2
IF(INOBJ(IB2).EQ.2)
F
T
B
B1
IPER=1



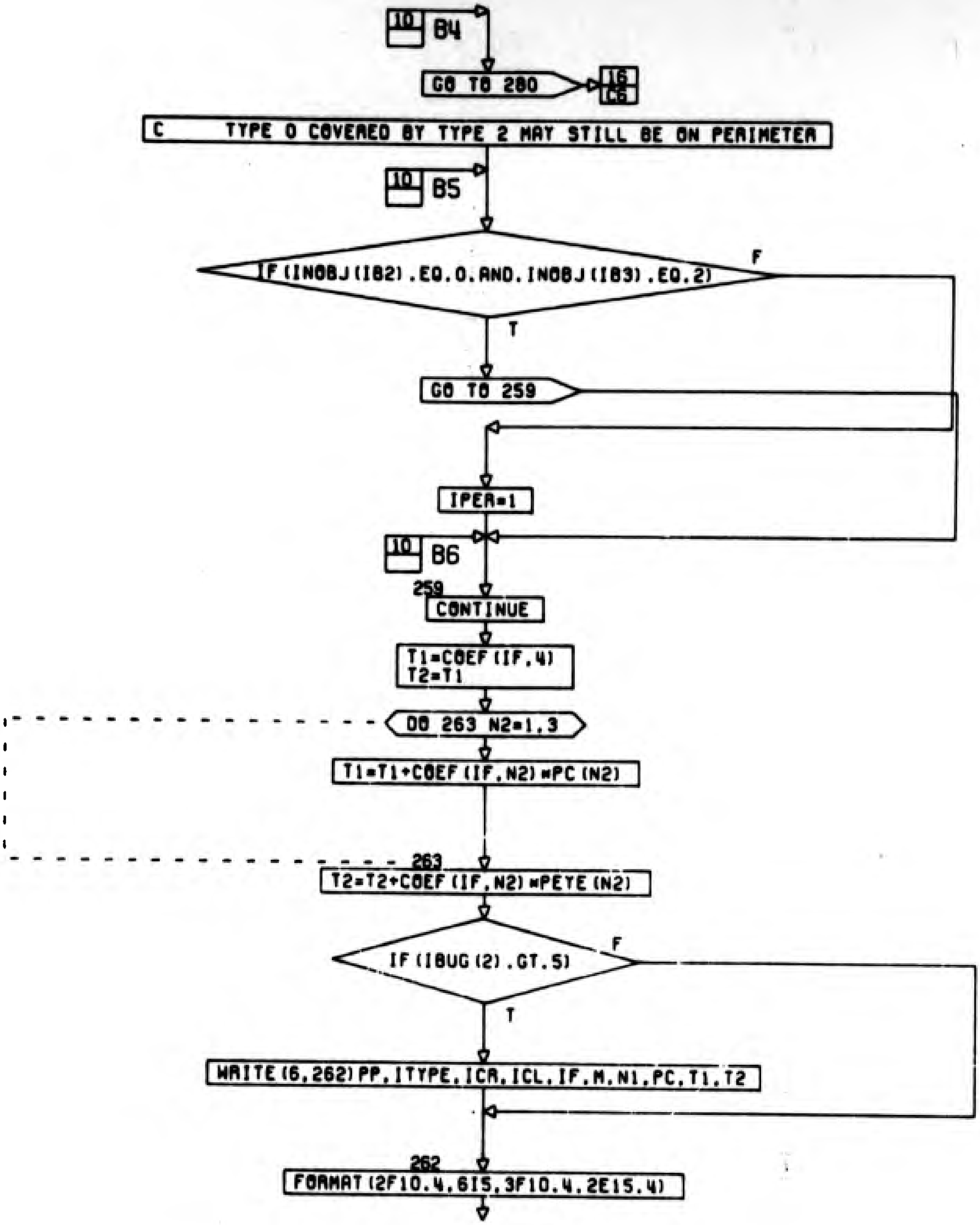
7
B1
DIS0=999.
DIS2=999.
IP=3
IF (N1.EQ.1)
F
T
IF (IPLOT) 275,280,275
≠
16
C5
0
16
C6
C SEE IF SAME POINT
DO 255 N2=1,2
IF (P(N1,N2).NE.P(N1-1,N2))
F
T
GO TO 257
255
CONTINUE
GO TO 280
16
C6
C CENTER OF SEGMENT
257
DO 258 N2=1,2
9

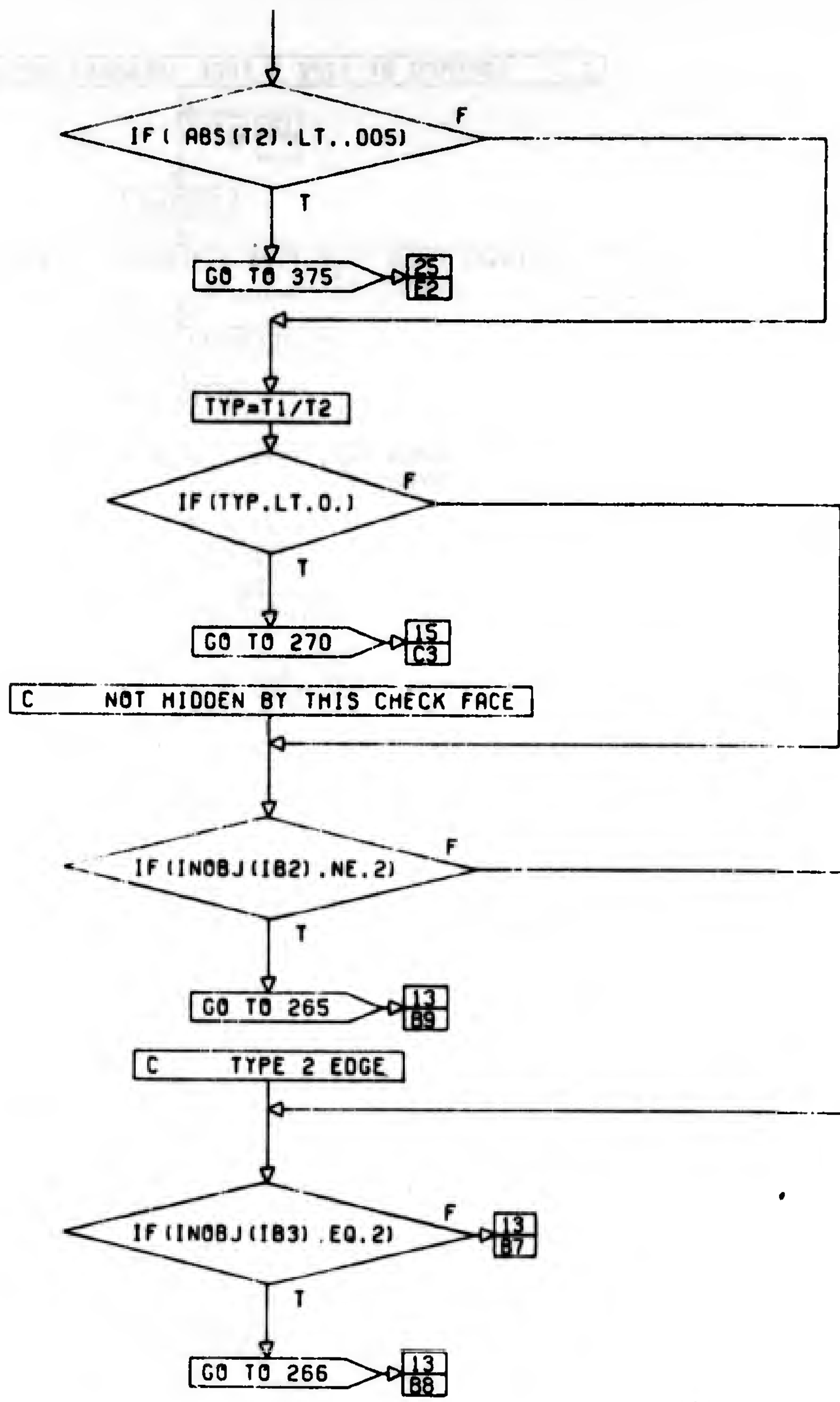


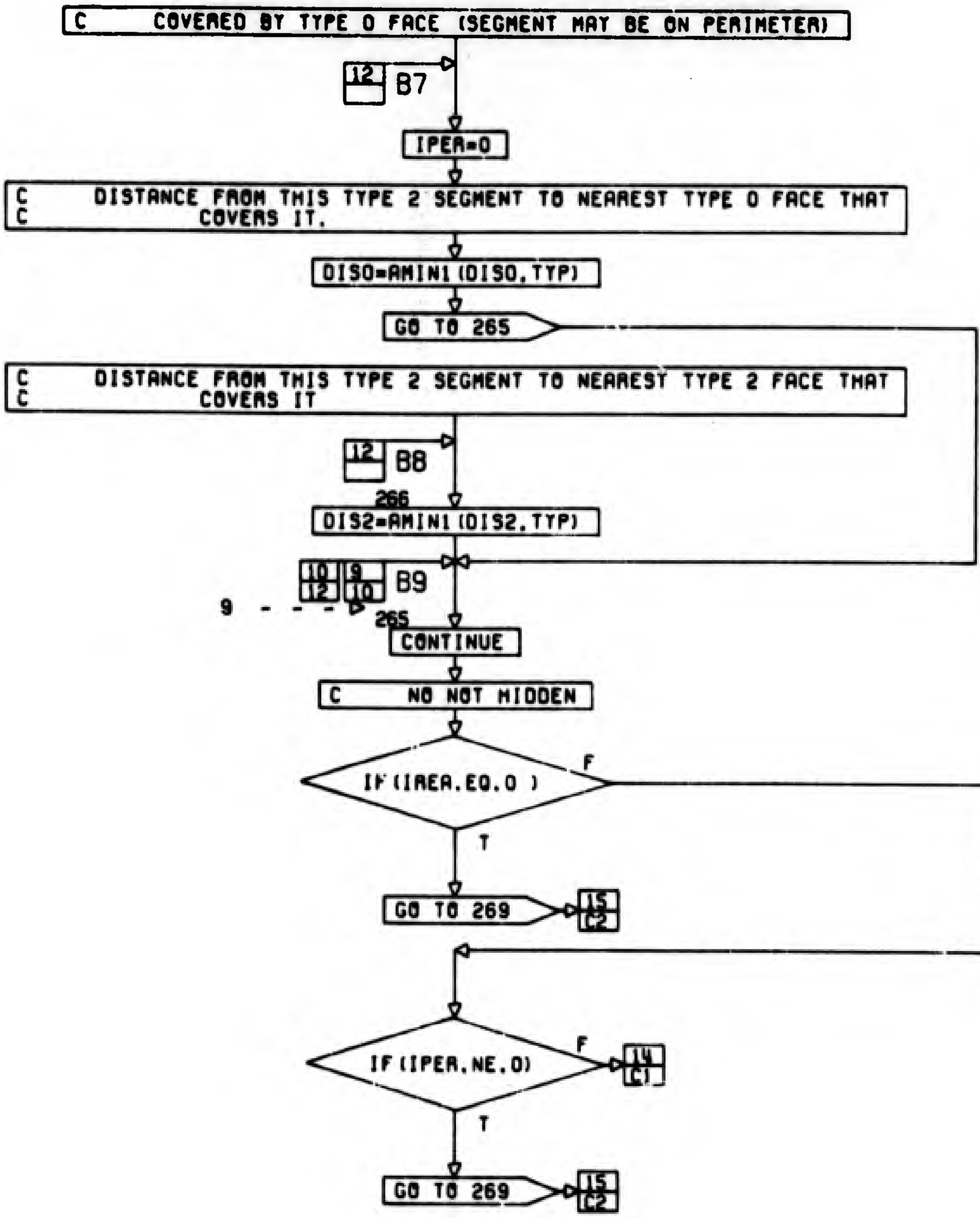
8
258
PP (N2) = (P (N1, N2) +P (N1-1, N2)) /2.
DO 260 N2=1,3
N3=1
IF (BL (1) .LE..01)
F
T
N3=2
260
PC (N2) =PVXTZ (K3, N2) +AL (N2) * (PP (N3) -P (1, N3)) /BL (N3)
C IS IT HIDDEN BY A FACE
DO 265 IF=1,LL
13
C CHECK FACE FOR TRANSPARENCY
IB3=LFCI (IF)
IF (INOBJ (IB3) .EQ.1.OR. INOBJ (IB3) .EQ.3)
F
T
GO TO 265
13
B9
C OPAQUE (WILL HIDE WHAT IS BEHIND IT)
C SKIP FACES ON EDGE
IF (JEDGE (M,3) .EQ.IF .OR. JEDGE (M,4) .EQ.IF)
F
10
B3
T
10
B2

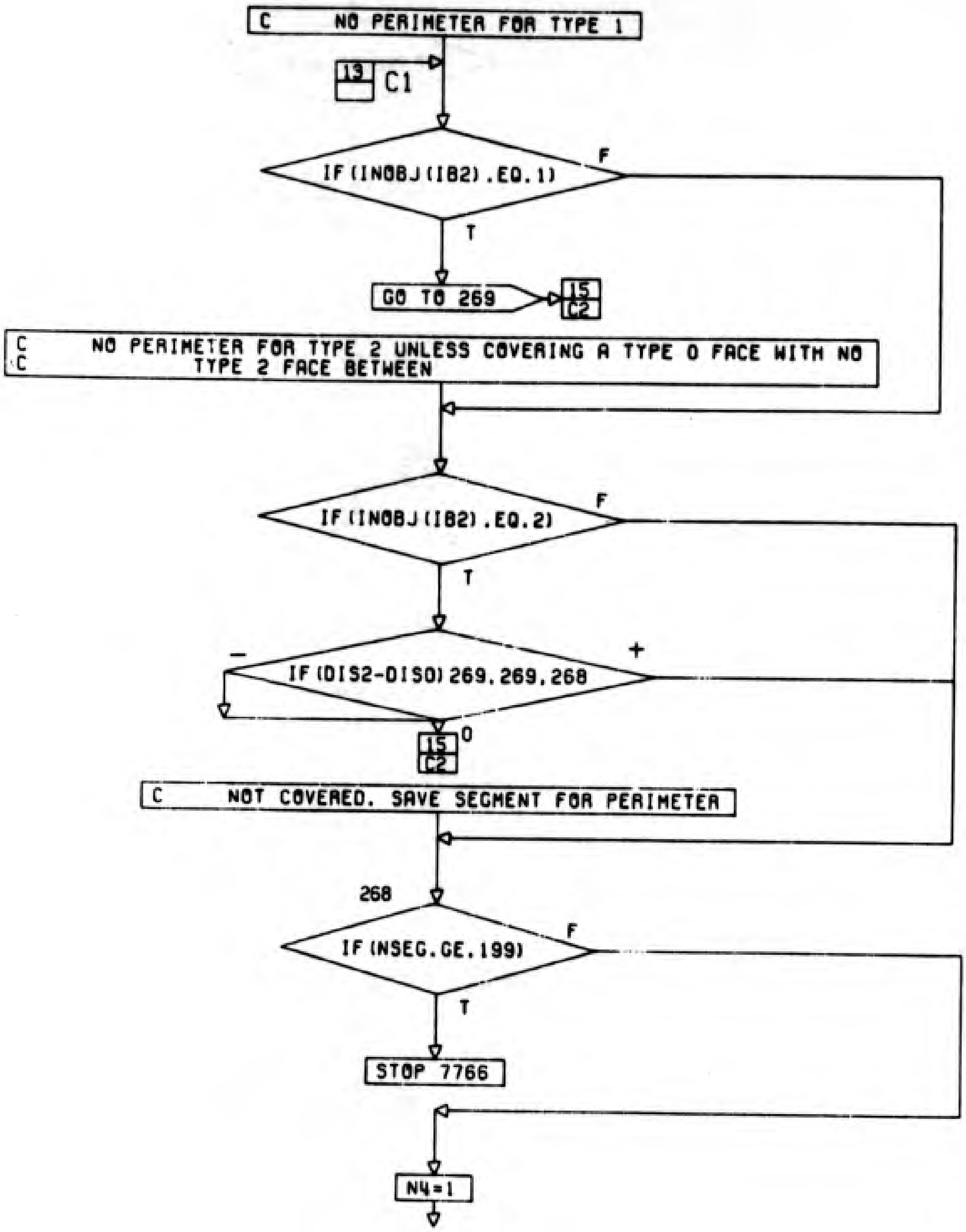


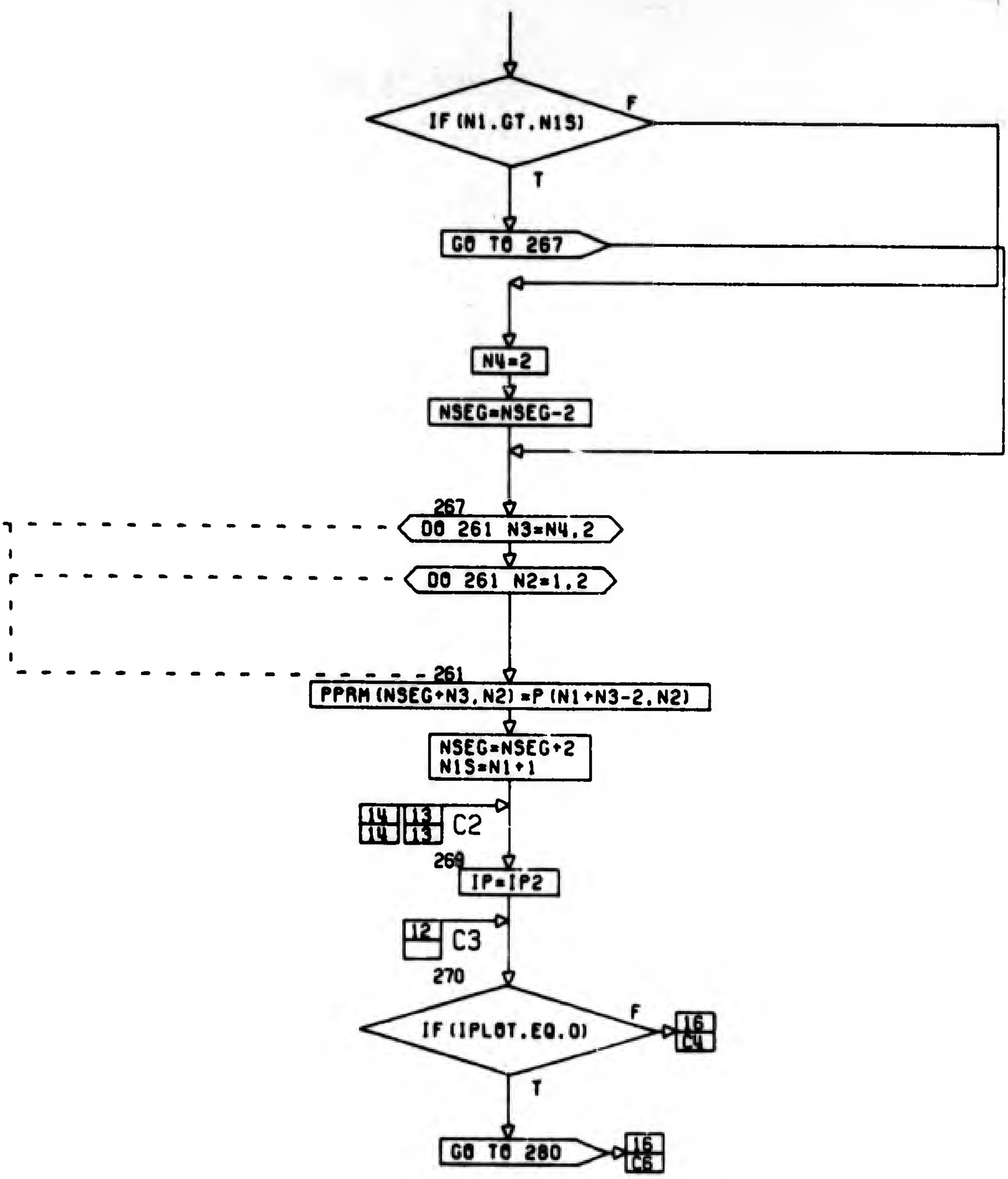
9
B2
GO TO 265
13
B9
9
B3
CALL COVER(IF,PP,PVRT,600)
IF(IBUG(2).GT.6)
F
T
WRITE(6,262)PP,ITYPE,ICR,ICL,IF,IB2,IB3
IF(ITYPE.EQ.0)
F
T
GO TO 265
13
B9
C COVERED
C INVERT PERIMETER, IF TYPE 2
IF(INOBJ(IB2).EQ.2)
F
T
GO TO 259
11
B6
IF(IPLOT.EQ.0)
F
11
B5
T
11
B4

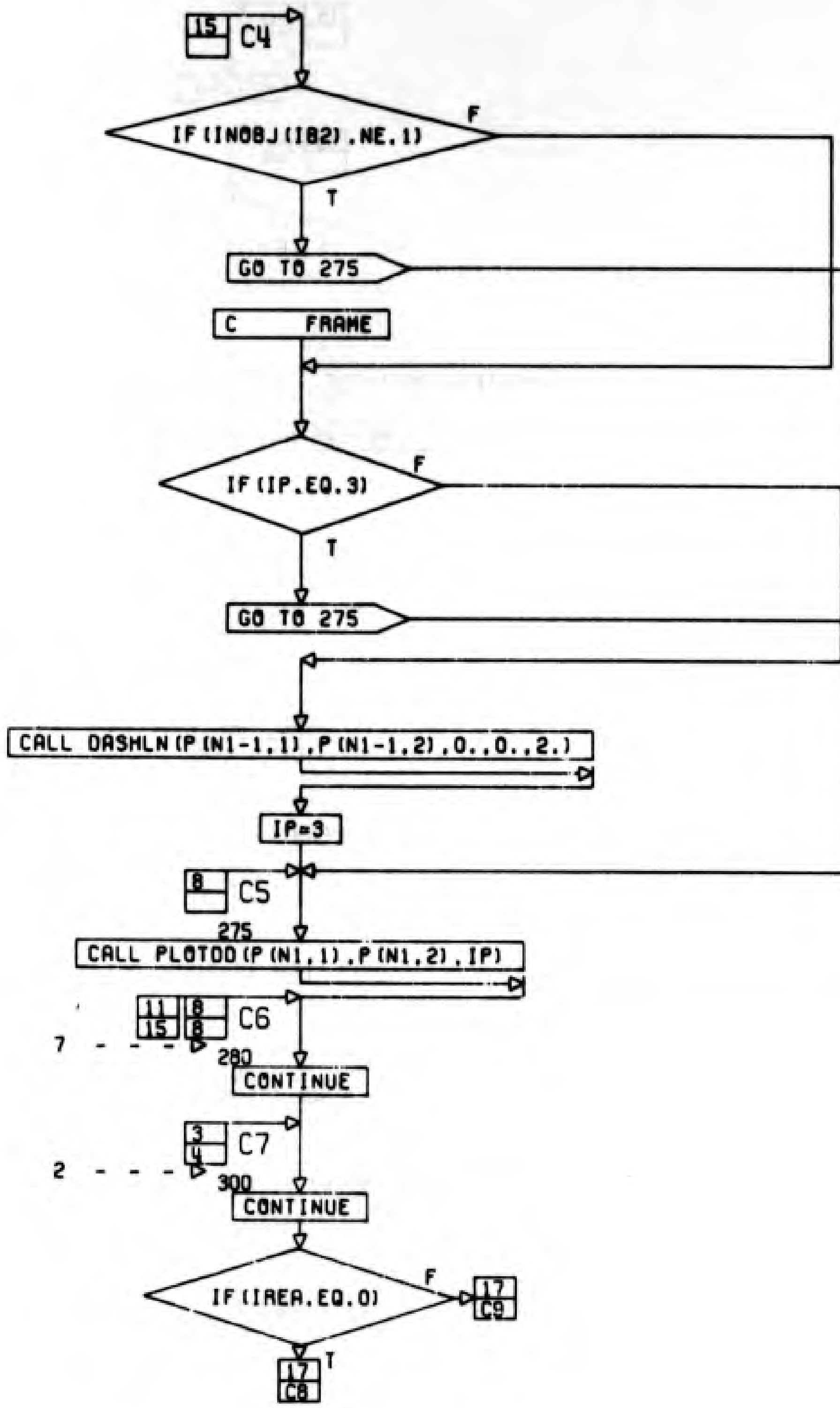

10
B4
GO TO 280
16
C6
C TYPE 0 COVERED BY TYPE 2 MAY STILL BE ON PERIMETER
10
B5
IF (INOBJ(IB2).EQ.0.AND.INOBJ(IB3).EQ.2)
F
T
GO TO 259
IPER=1
10
B6
259
CONTINUE
T1=COEF(IF,4)
T2=T1
DO 263 N2=1,3
T1=T1+COEF(IF,N2)*PC(N2)
263
T2=T2+COEF(IF,N2)*PEYE(N2)
IF(IBUG(2).GT.5)
F
T
WRITE(6,262)PP,ITYPE,ICR,ICL,IF,M,N1,PC,T1,T2
262
FORMAT(2F10.4,6I5,3F10.4,2E15.4)

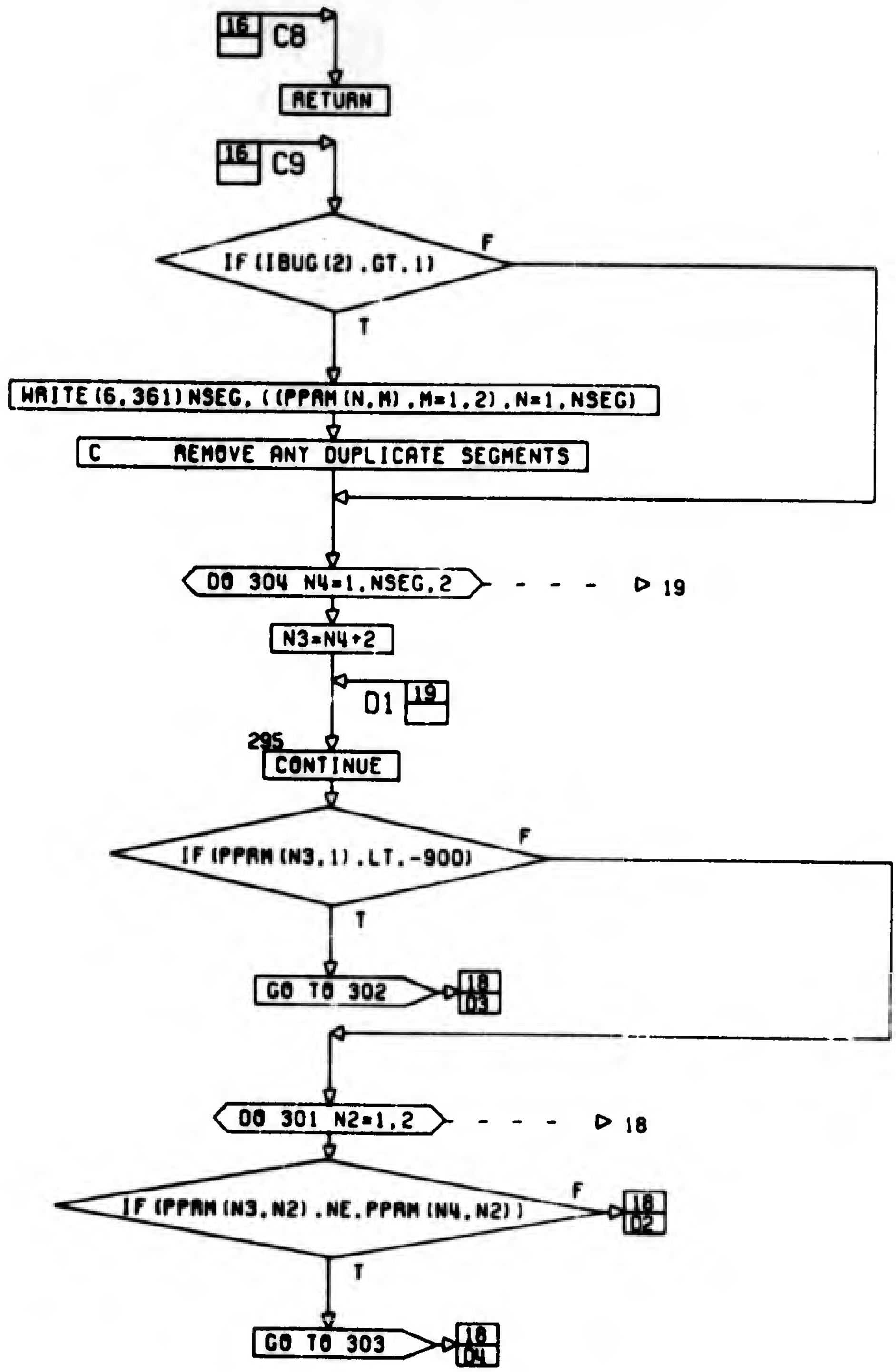

IF (ABS(T2).LT..005)
F
T
GO TO 375
25
E2
TYP=T1/T2
IF(TYP.LT.0.)
F
T
GO TO 270
15
C3
C NOT HIDDEN BY THIS CHECK FACE
IF(INOBJ(IB2).NE.2)
F
T
GO TO 265
13
B9
C TYPE 2 EDGE
IF(INOBJ(IB3).EQ.2)
F
13
B7
T
GO TO 266
13
B8


C COVERED BY TYPE 0 FACE (SEGMENT MAY BE ON PERIMETER)
12
B7
IPER=0
C DISTANCE FROM THIS TYPE 2 SEGMENT TO NEAREST TYPE 0 FACE THAT
C COVERS IT.
DIS0=AMIN1(DIS0,TYP)
GO TO 265
C DISTANCE FROM THIS TYPE 2 SEGMENT TO NEAREST TYPE 2 FACE THAT
C COVERS IT
12
B8
266
DIS2=AMIN1(DIS2,TYP)
10
12
9
10
B9
9
265
CONTINUE
C NO NOT HIDDEN
IF(IREA.EQ.0)
F
T
GO TO 269
15
C2
IF(IPER.NE.0)
F
14
C1
T
GO TO 269
15
C2

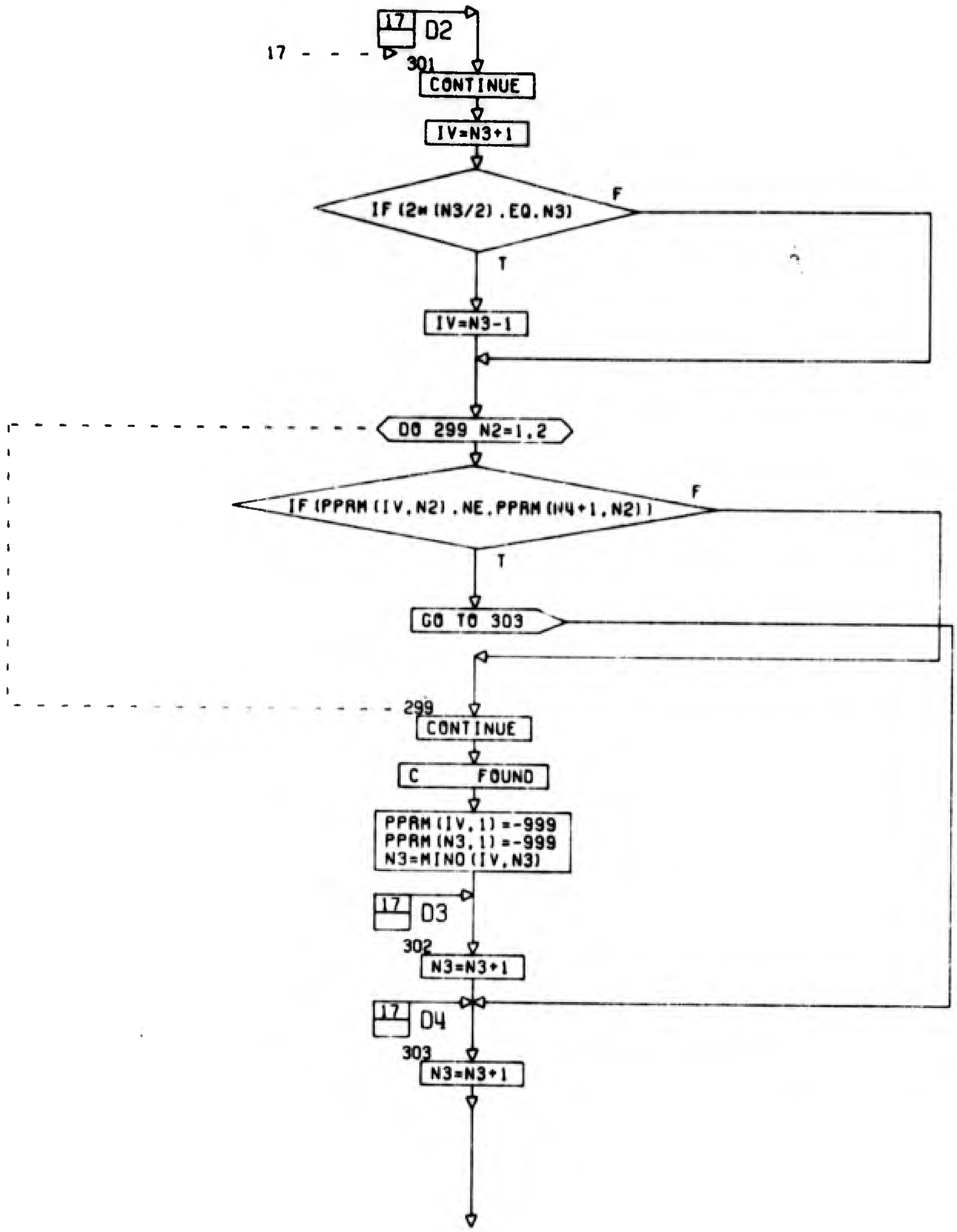

C NO PERIMETER FOR TYPE 1
13
C1
IF(INOBJ(IB2).EQ.1)
F
T
GO TO 269
15
C2
C NO PERIMETER FOR TYPE 2 UNLESS COVERING A TYPE 0 FACE WITH NO
C TYPE 2 FACE BETWEEN
IF(INOBJ(IB2).EQ.2)
F
T
IF(DIS2-DIS0) 269,269,268
-
+
0
15
C2
C NOT COVERED. SAVE SEGMENT FOR PERIMETER
268
IF(NSEG.GE.199)
F
T
STOP 7766
N4=1

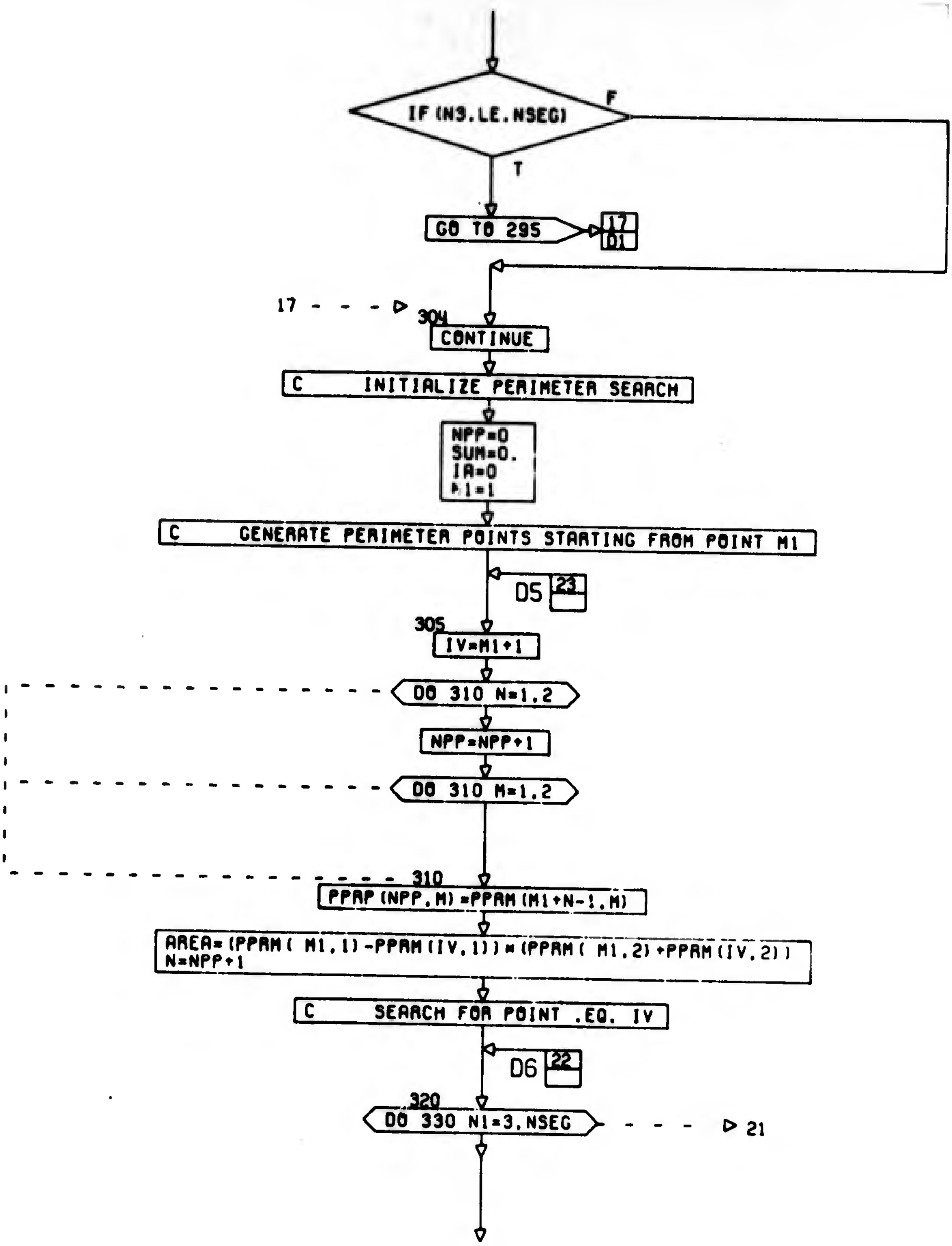

IF (N1.GT.N1S)
F
T
GO TO 267
N4=2
NSEG=NSEG-2
267
DO 261 N3=N4,2
DO 261 N2=1,2
261
PPRM (NSEG+N3, N2) =P (N1+N3-2, N2)
NSEG=NSEG+2
N1S=N1+1
14
14
13
13
C2
260
IP=IP2
12
C3
270
IF (IPLOT.EQ.0)
F
16
C4
T
GO TO 280
16
C6

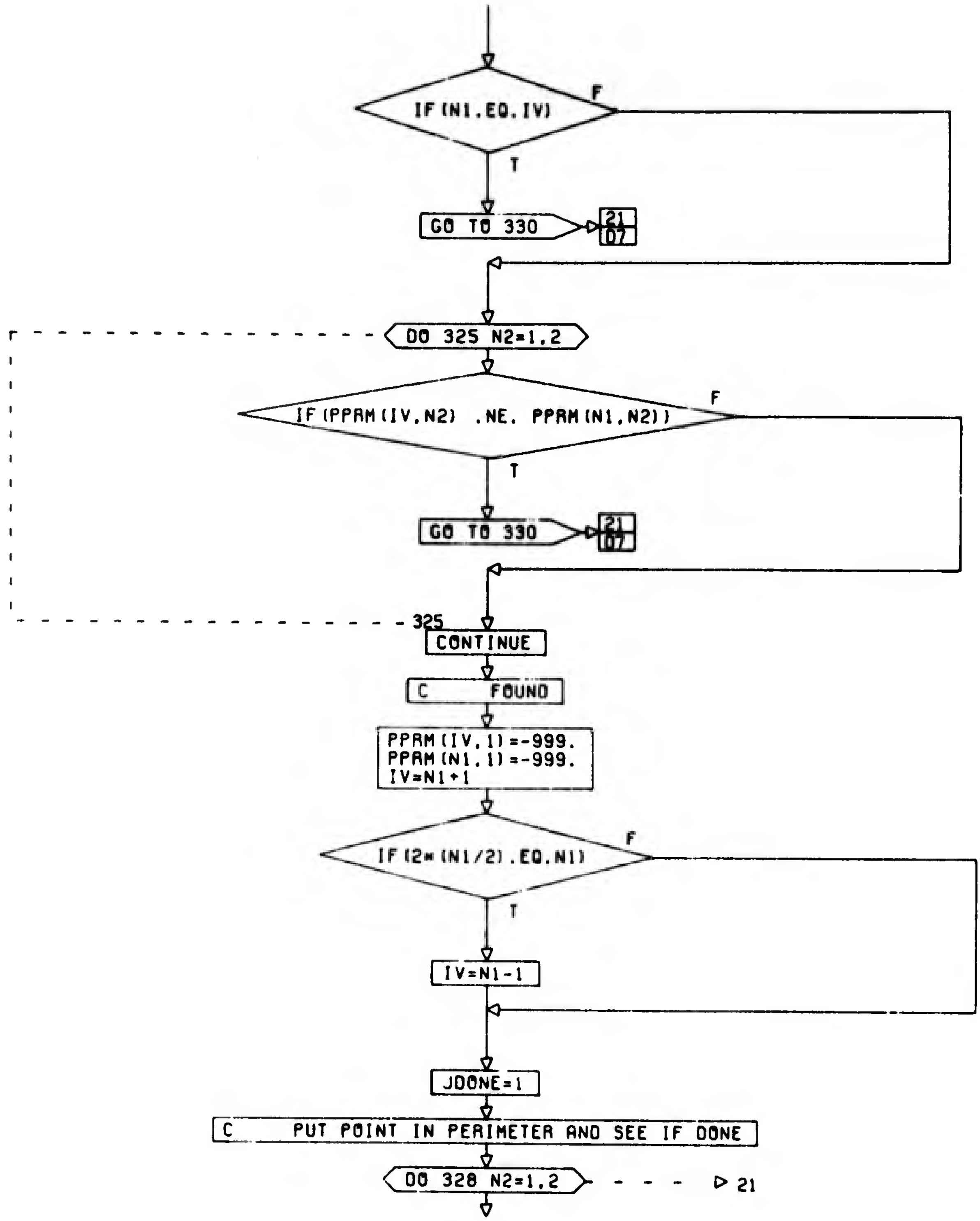

15
C4
IF(INOBJ(IB2).NE.1)
F
T
GO TO 275
C FRAME
IF(IP.EQ.3)
F
T
GO TO 275
CALL DASHLN(P(N1-1,1),P(N1-1,2),0.,0.,2.)
IP=3
8
C5
275
CALL PLOTOD(P(N1,1),P(N1,2),IP)
11 8
15 8
C6
7
280
CONTINUE
3
4
C7
2
300
CONTINUE
IF(IAREA.EQ.0)
F
17
C9
T
17
C8

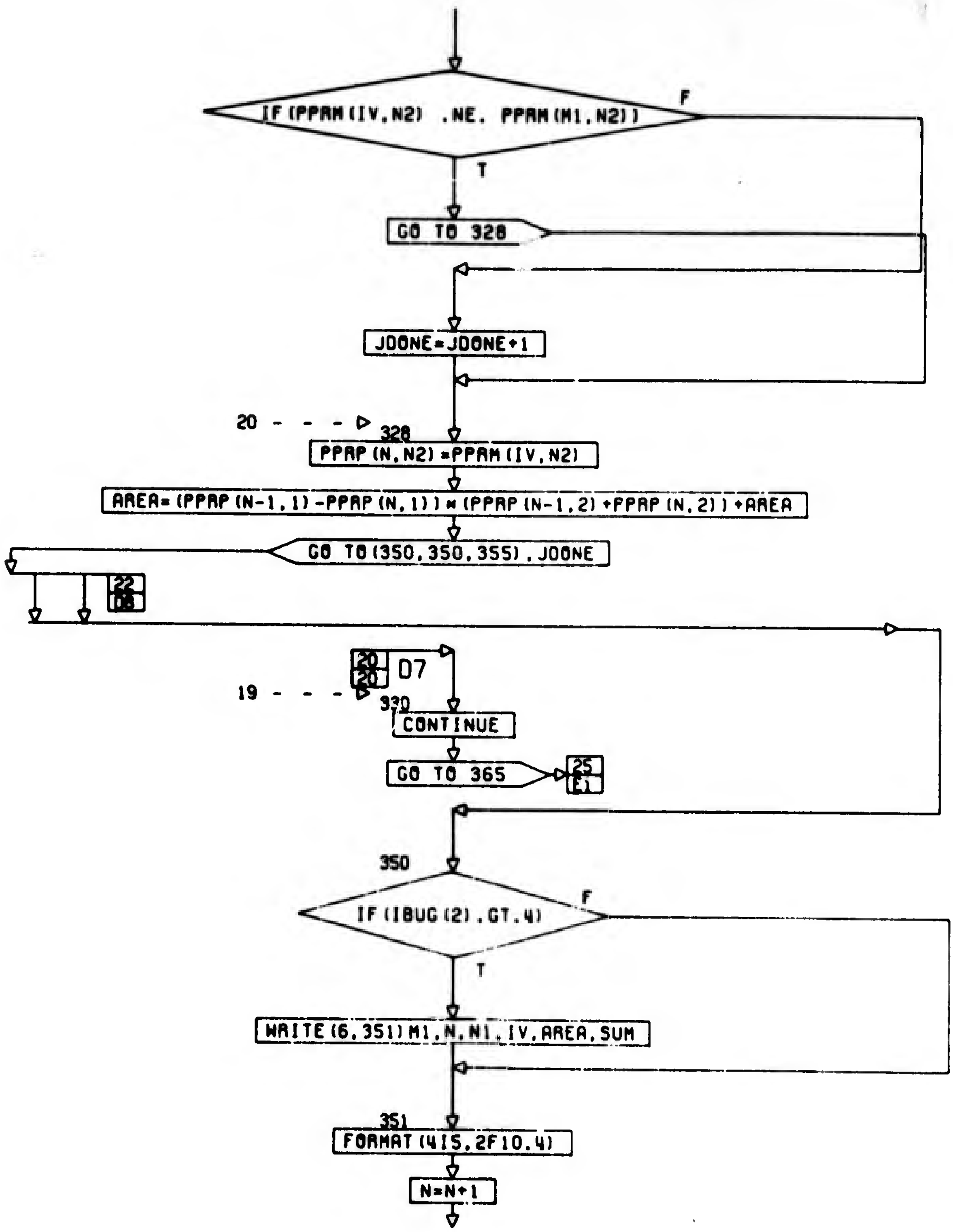


16 C8
RETURN
16 C9
IF (IBUG(2).GT.1)
F
T
WRITE(6,361)NSEG,((PPRM(N,M),M=1,2),N=1,NSEG)
C REMOVE ANY DUPLICATE SEGMENTS
DO 304 N4=1,NSEG,2
19
N3=N4+2
D1 19
295
CONTINUE
IF (PPRM(N3,1).LT.-900)
F
T
GO TO 302
18 03
DO 301 N2=1,2
18
IF (PPRM(N3,N2).NE.PPRM(N4,N2))
F
18 02
T
GO TO 303
18 04

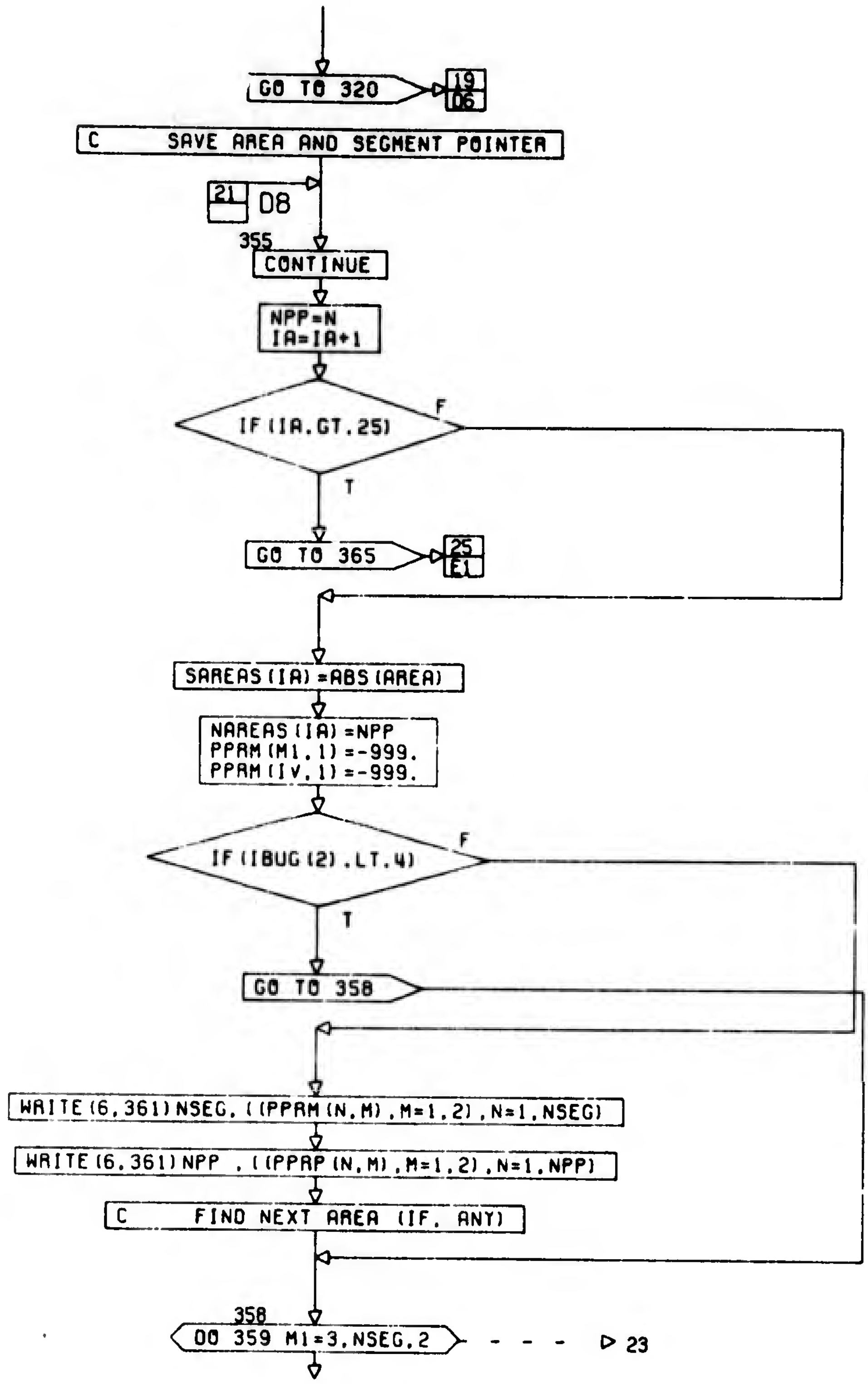

17
D2
17
301
CONTINUE
IV=N3+1
IF (2*(N3/2).EQ.N3)
F
T
IV=N3-1
DO 299 N2=1,2
IF (PPRM(IV,N2).NE.PPRM(N4+1,N2))
F
T
GO TO 303
299
CONTINUE
C FOUND
PPRM(IV,1)=-999
PPRM(N3,1)=-999
N3=MINO(IV,N3)
17
D3
302
N3=N3+1
17
D4
303
N3=N3+1

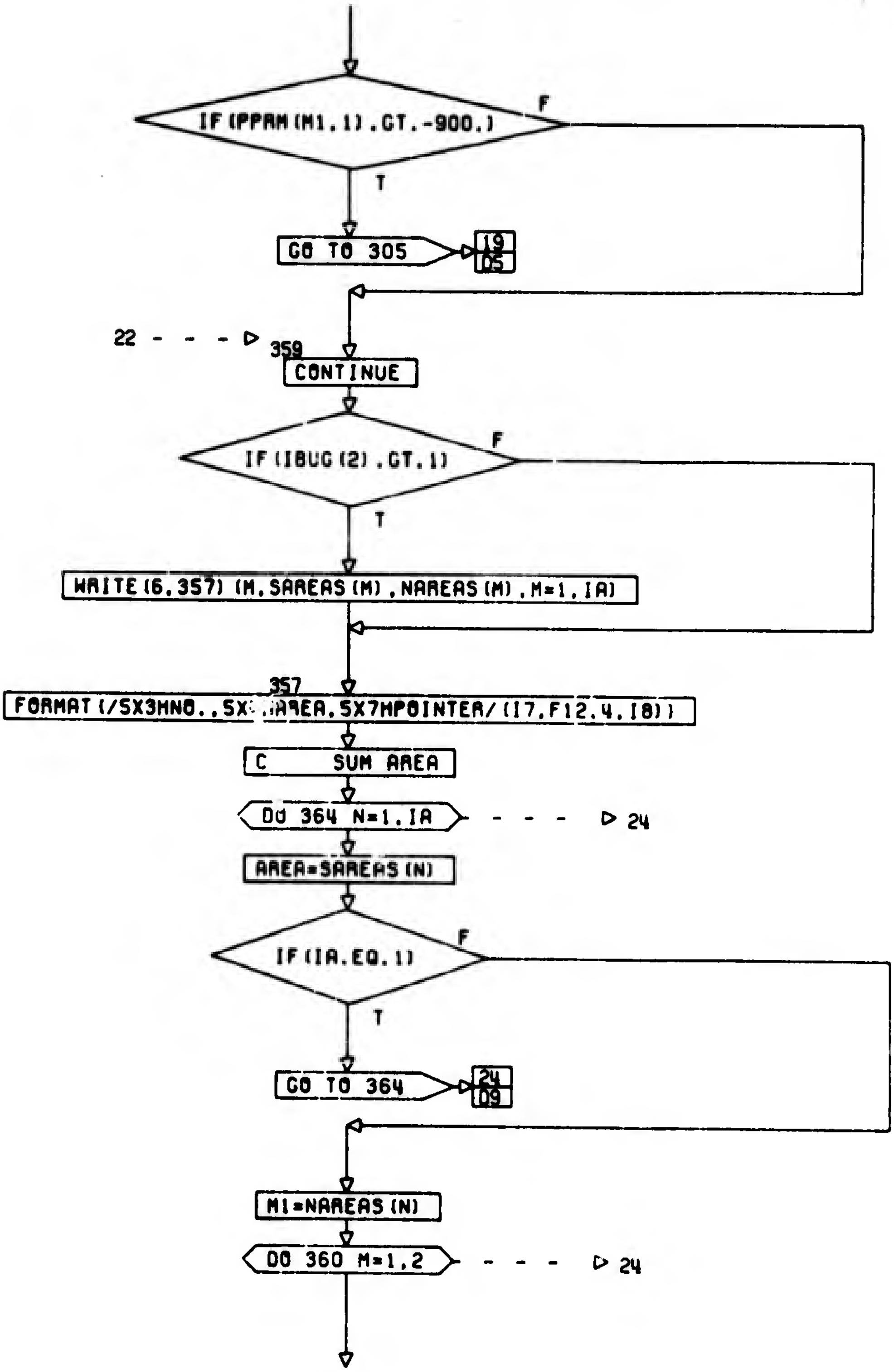

IF (N3.LE.NSEG)
F
T
GO TO 295
17
D1
17
304
CONTINUE
C INITIALIZE PERIMETER SEARCH
NPP=0
SUM=0.
IA=0
M1=1
C GENERATE PERIMETER POINTS STARTING FROM POINT M1
D5
23
305
IV=M1+1
DO 310 N=1,2
NPP=NPP+1
DO 310 M=1,2
310
PPAP(NPP,M)=PPAM(M1+N-1,M)
AREA=(PPAM(M1,1)-PPAM(IV,1))*(PPAM(M1,2)+PPAM(IV,2))
N=NPP+1
C SEARCH FOR POINT .EQ. IV
D6
22
320
DO 330 N1=3,NSEG
21

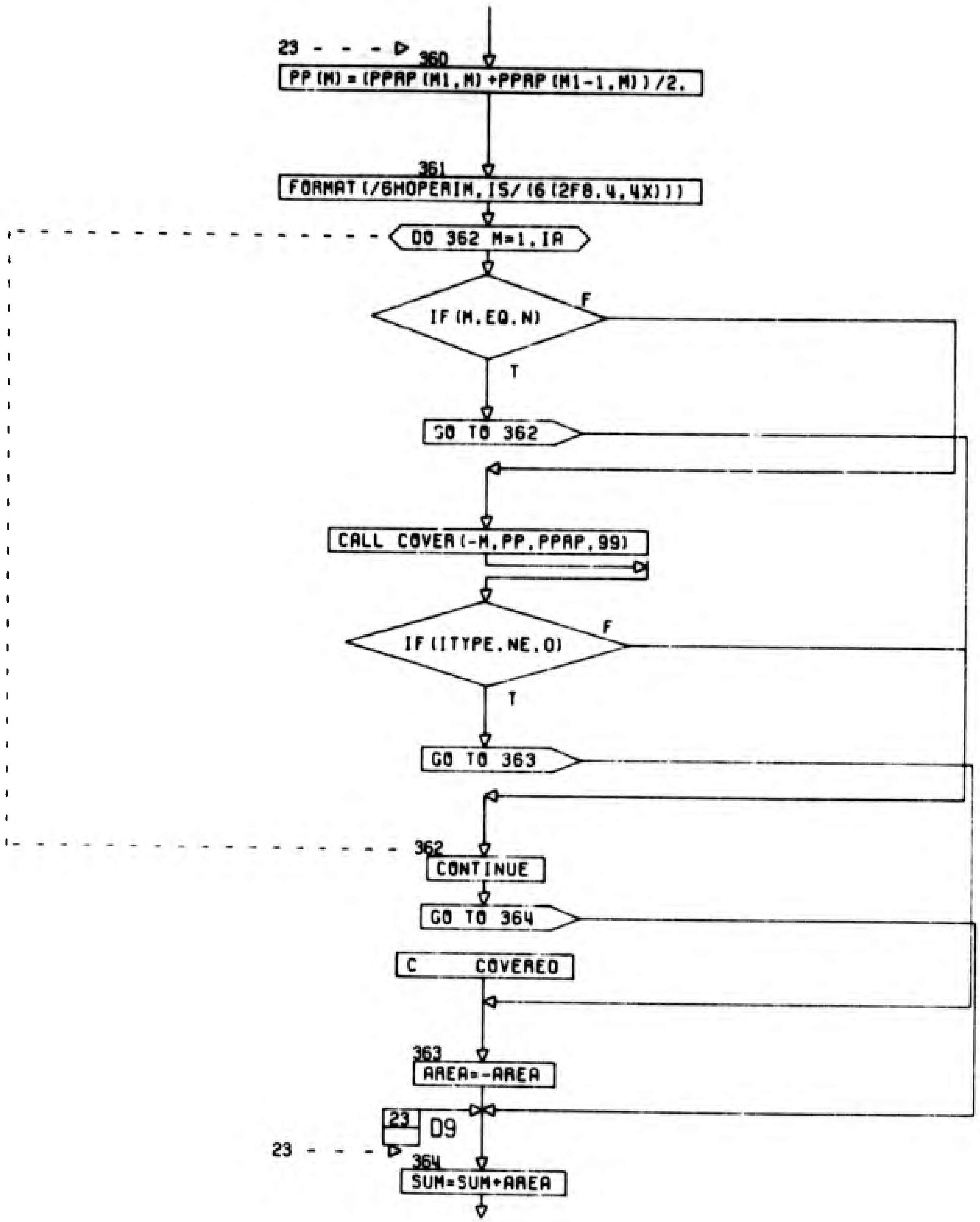

IF (N1.EQ.IV)
F
T
GO TO 330
21
07
DO 325 N2=1,2
IF (PPRM(IV,N2) .NE. PPRM(N1,N2))
F
T
GO TO 330
21
07
325
CONTINUE
C FOUND
PPRM(IV,1)=-999.
PPRM(N1,1)=-999.
IV=N1+1
IF (2*(N1/2).EQ.N1)
F
T
IV=N1-1
JDONE=1
C PUT POINT IN PERIMETER AND SEE IF DONE
DO 328 N2=1,2
21

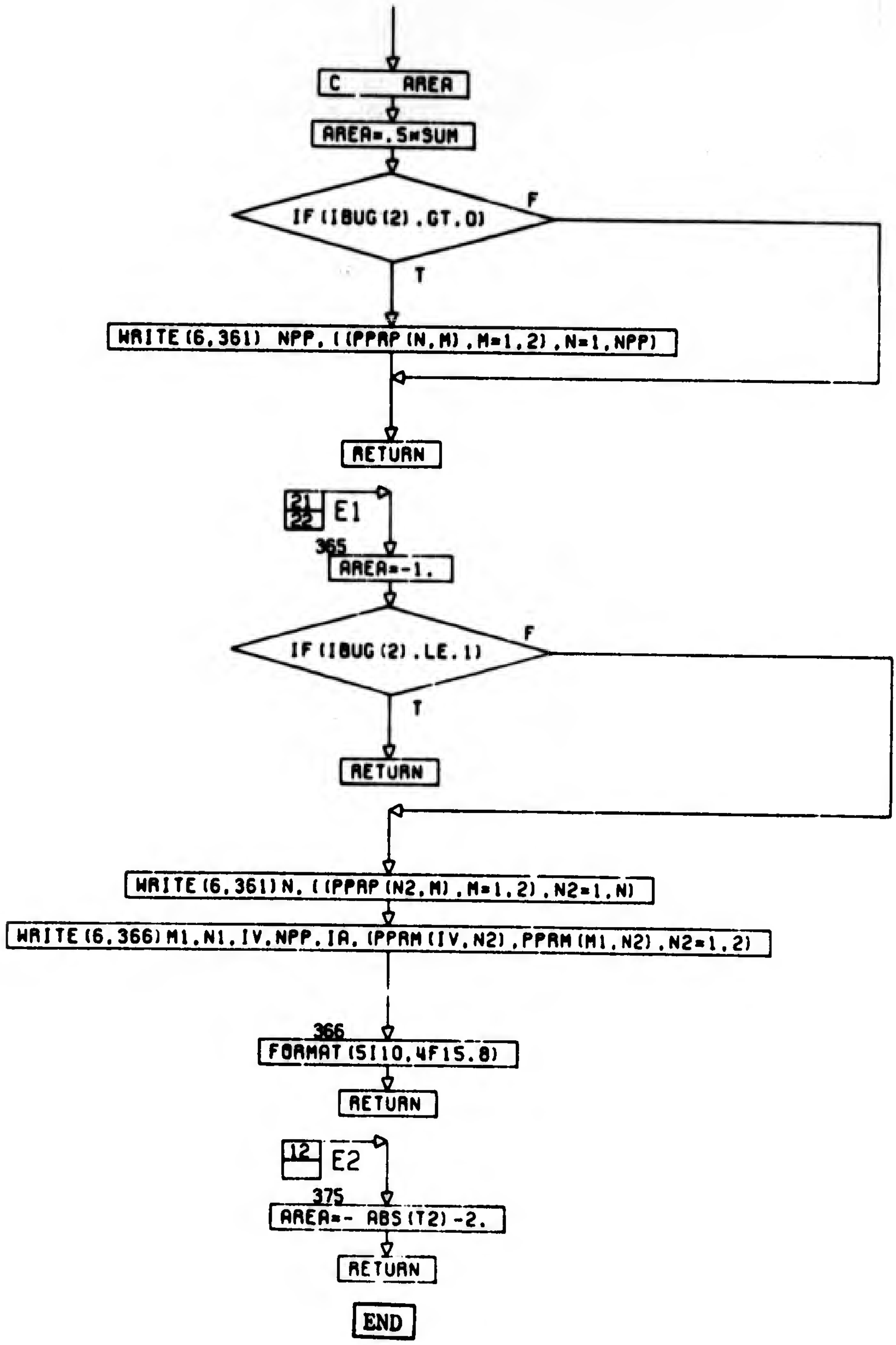


IF (PPRM(IV,N2) .NE. PPRM(M1,N2))
F
T
GO TO 328
JDONE=JDONE+1
20
328
PPRP(N,N2)=PPRM(IV,N2)
AREA=(PPRP(N-1,1)-PPRP(N,1))*(PPRP(N-1,2)+PPRP(N,2))+AREA
GO TO (350,350,355),JDONE
22
08
20
20
07
19
330
CONTINUE
GO TO 365
25
E1
350
IF (IBUG(2).GT.4)
F
T
WRITE(6,351)M1,N,N1,IV,AREA,SUM
351
FORMAT(4I5,2F10.4)
N=N+1

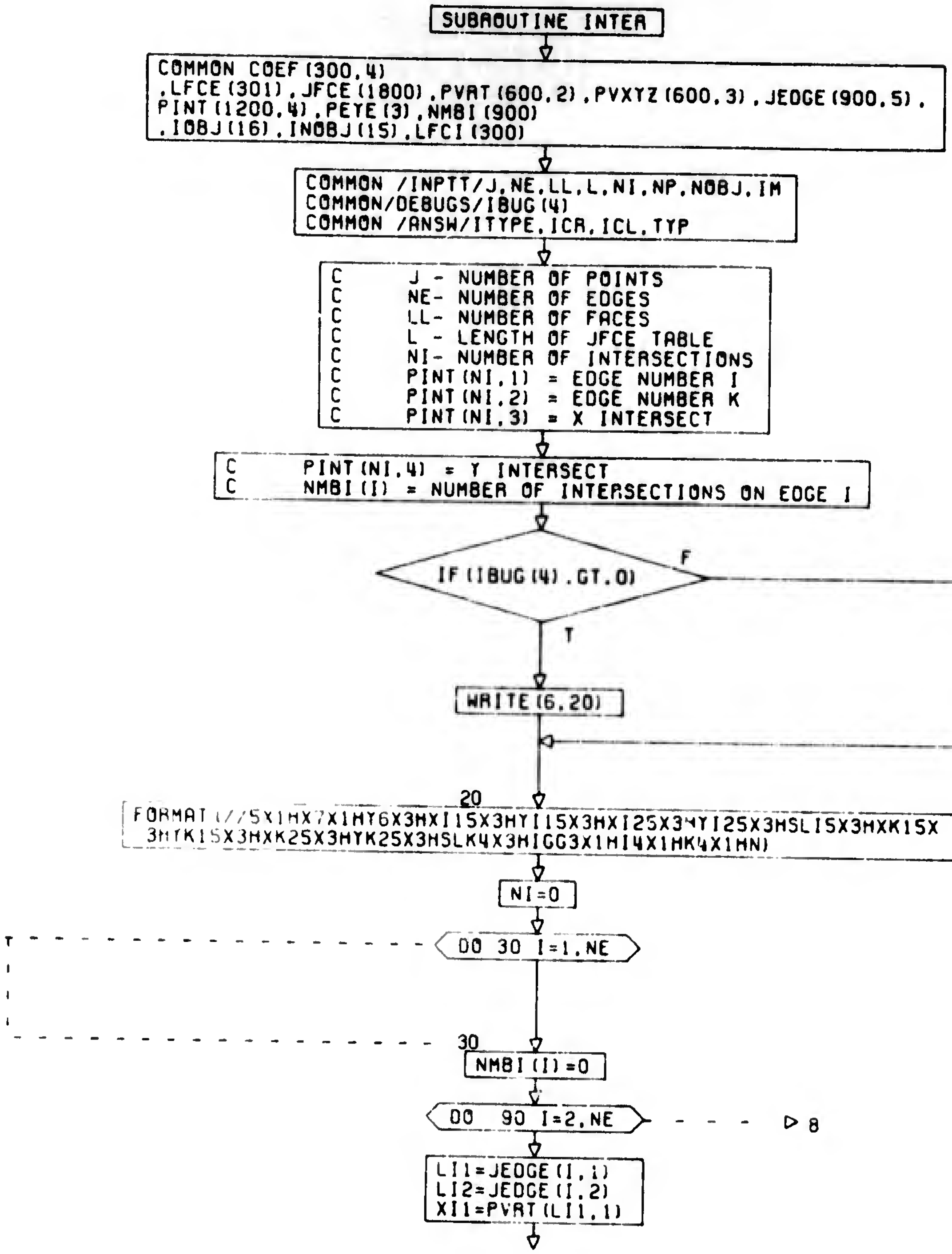

GO TO 320
19
D6
C SAVE AREA AND SEGMENT POINTER
21
D8
355
CONTINUE
NPP=N
IA=IA+1
IF (IA.GT.25)
F
T
GO TO 365
25
E1
SAREAS(IA)=ABS(AREA)
NAREAS(IA)=NPP
PPAM(M1,1)=-999.
PPAM(IV,1)=-999.
IF (IBUG(2).LT.4)
F
T
GO TO 358
WRITE(6,361)NSEG,((PPAM(N,M),M=1,2),N=1,NSEG)
WRITE(6,361)NPP ,((PPAP(N,M),M=1,2),N=1,NPP)
C FIND NEXT AREA (IF, ANY)
358
DO 359 M1=3,NSEG,2
23

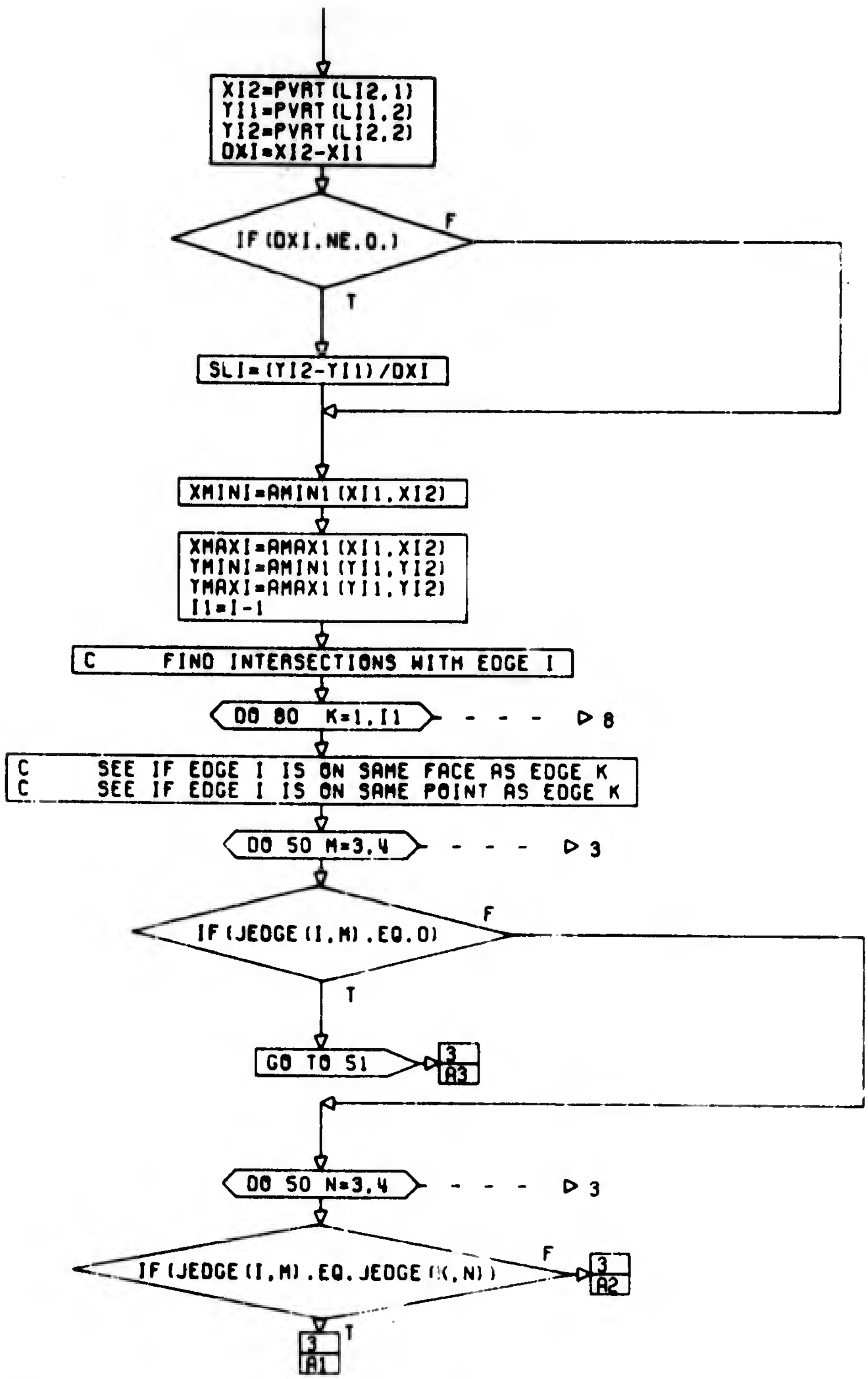

IF (PPAM (M1, 1) .GT. -900.)
F
T
GO TO 305
19
05
22
359
CONTINUE
IF (IBUG (2) .GT. 1)
F
T
WRITE (6, 357) (M, SAREAS (M), NAREAS (M), M=1, IA)
357
FORMAT (/5X3HNO., 5X
AREA, 5X7HPOINTER/ (I7, F12.4, I8))
C SUM AREA
DO 364 N=1, IA
24
AREA=SAREAS (N)
IF (IA.EQ.1)
F
T
GO TO 364
24
09
M1=NAREAS (N)
DO 360 M=1, 2
24


23
360
PP(M)=(PPAP(M1,M)+PPAP(M1-1,M))/2.
361
FORMAT(/6HOPERIM,I5/(6(2F8.4,4X)))
DO 362 M=1,IA
IF(M.EQ.N)
F
T
GO TO 362
CALL COVER(-M,PP,PPAP,99)
IF(ITYPE.NE.0)
F
T
GO TO 363
362
CONTINUE
GO TO 364
C COVERED
363
AREA=-AREA
23
09
23
364
SUM=SUM+AREA

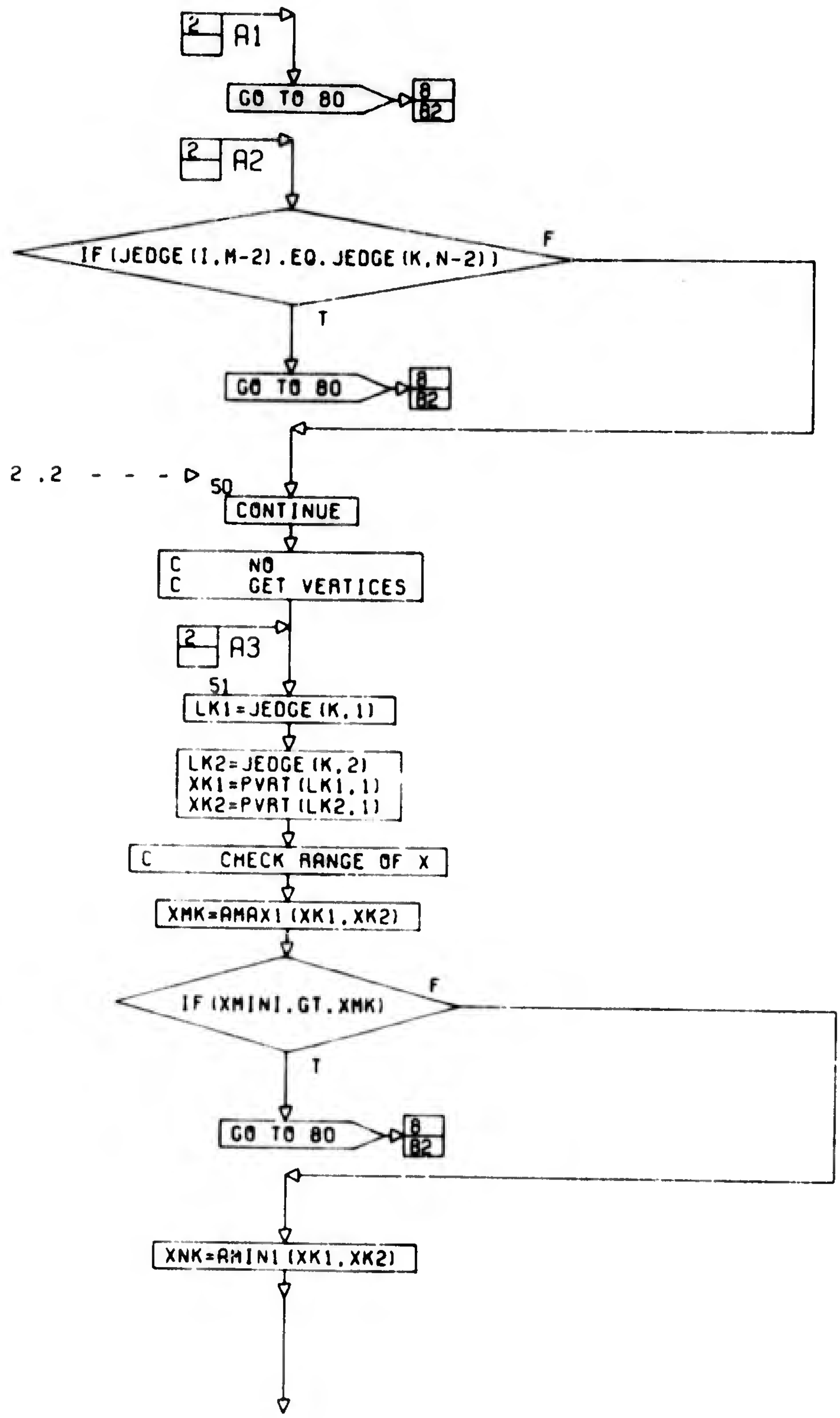

C AREA
AREA=.5*SUM
IF(IBUG(2).GT.0)
F
T
WRITE(6,361) NPP,((PPRP(N,M),M=1,2),N=1,NPP)
RETURN
21
22
E1
365
AREA=-1.
IF(IBUG(2).LE.1)
F
T
RETURN
WRITE(6,361)N,((PPRP(N2,M),M=1,2),N2=1,N)
WRITE(6,366)M1,N1,IV,NPP,IA,(PPRM(IV,N2),PPRM(M1,N2),N2=1,2)
366
FORMAT(5I10,4F15.8)
RETURN
12
E2
375
AREA=- ABS(T2)-2.
RETURN
END

SUBROUTINE INTER

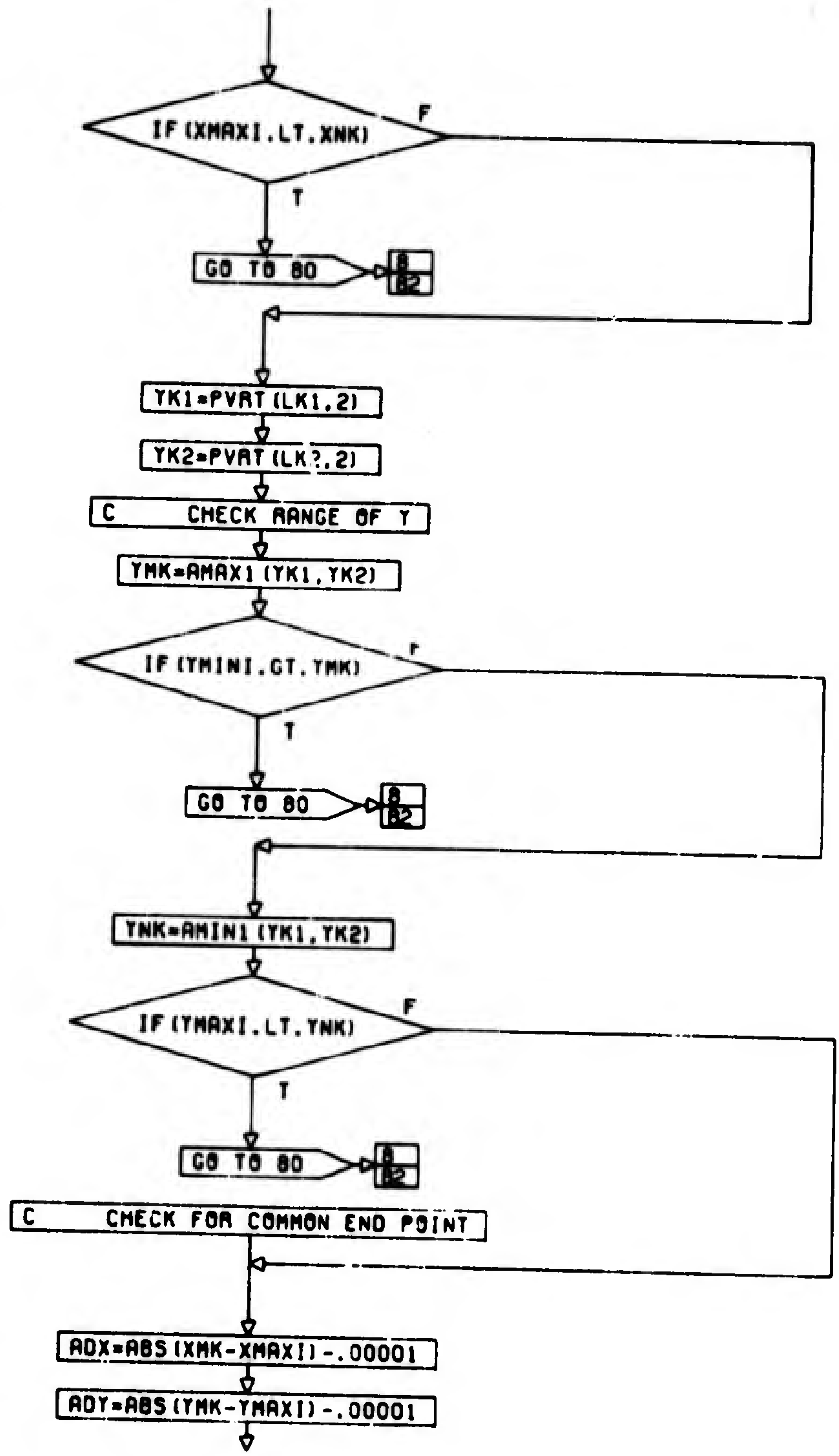


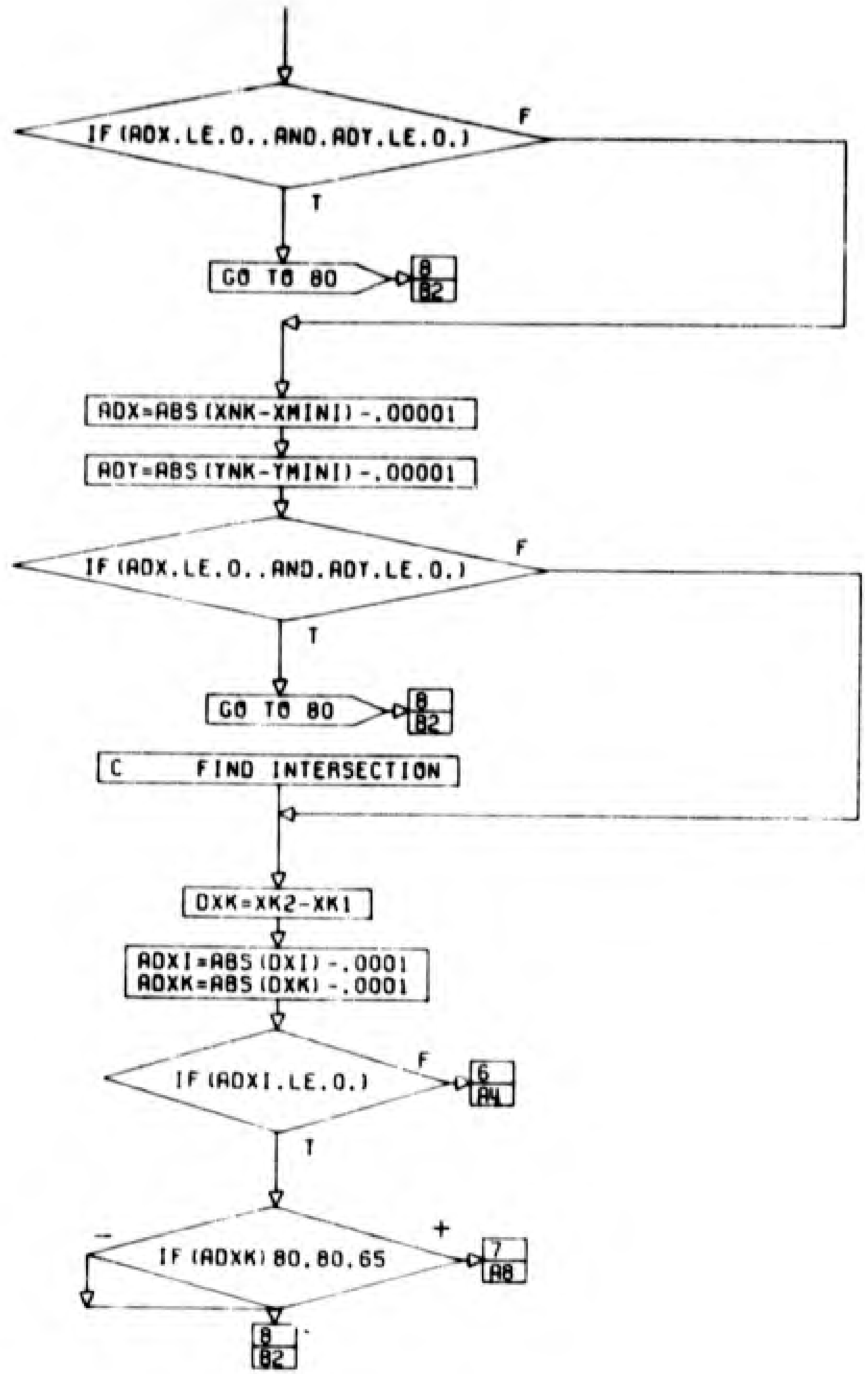


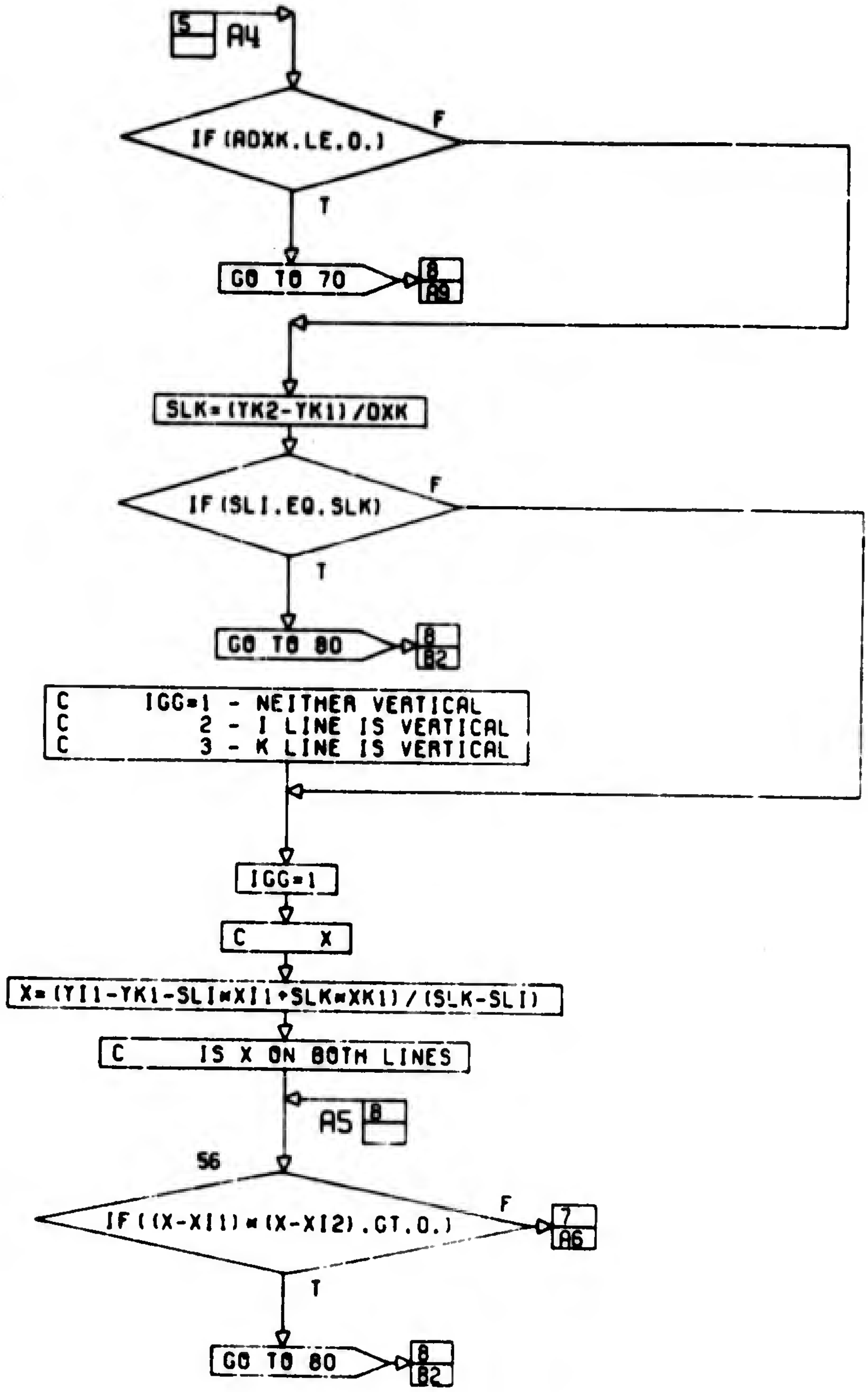
XI2=PVRT(LI2,1)
YI1=PVRT(LI1,2)
YI2=PVRT(LI2,2)
DXI=XI2-XI1
IF(DXI.NE.0.)
F
T
SLI=(YI2-YI1)/DXI
XMINI=AMIN1(XI1,XI2)
XMAXI=AMAX1(XI1,XI2)
YMINI=AMIN1(YI1,YI2)
YMAXI=AMAX1(YI1,YI2)
I1=I-1
C FIND INTERSECTIONS WITH EDGE I
DO 80 K=1,I1
8
C SEE IF EDGE I IS ON SAME FACE AS EDGE K
C SEE IF EDGE I IS ON SAME POINT AS EDGE K
DO 50 M=3,4
3
IF(JEDGE(I,M).EQ.0)
F
T
GO TO 51
3
A3
DO 50 N=3,4
3
IF(JEDGE(I,M).EQ.JEDGE(K,N))
F
3
A2
T
3
A1



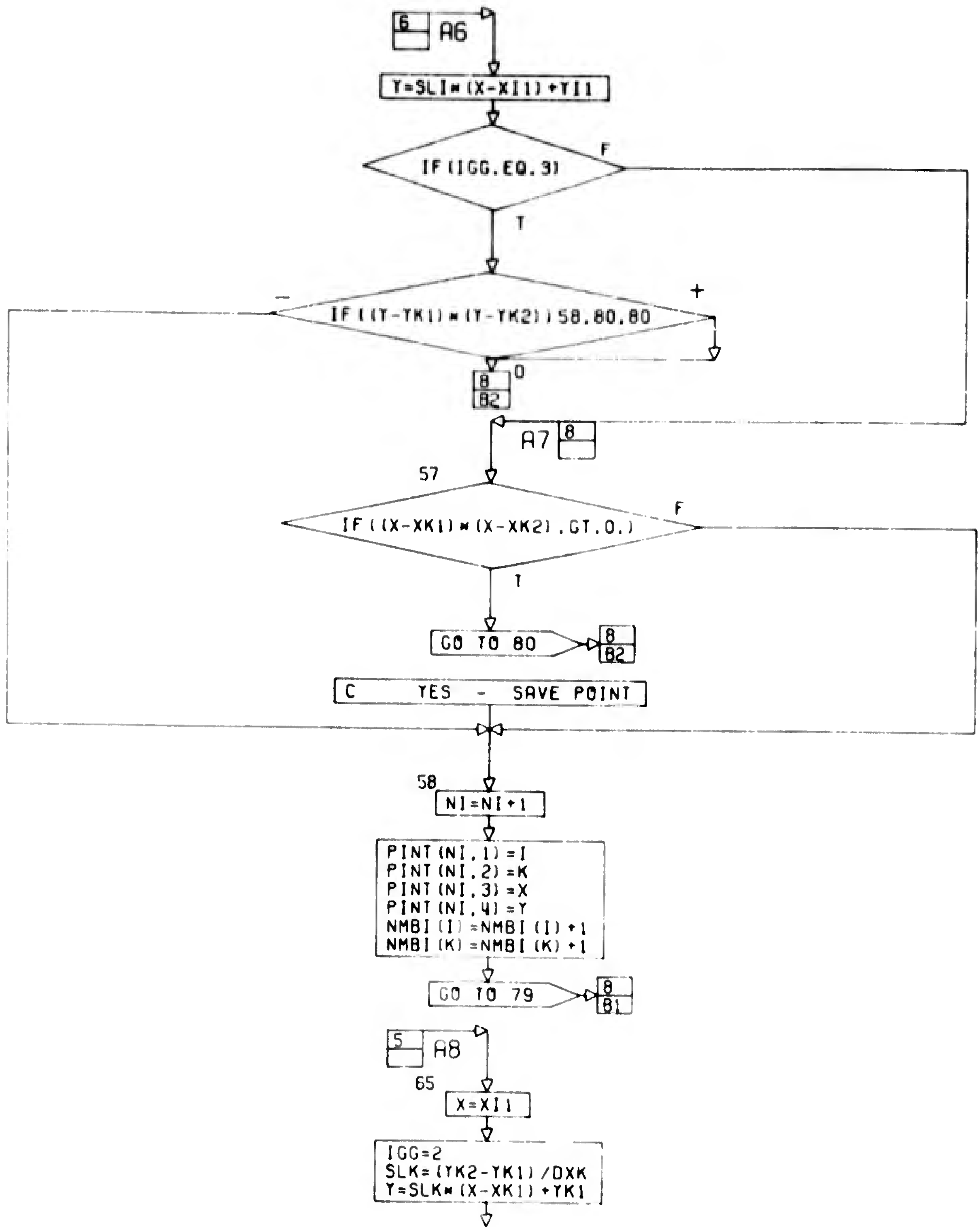
2 A1
GO TO 80
8 B2
2 A2
IF (JEDGE (I,M-2) .EQ. JEDGE (K,N-2))
F
T
GO TO 80
8 B2
2 .2
50
CONTINUE
C NO
C GET VERTICES
2 A3
51
LK1=JEDGE (K,1)
LK2=JEDGE (K,2)
XK1=PVAT (LK1,1)
XK2=PVAT (LK2,1)
C CHECK RANGE OF X
XMK=AMAX1 (XK1,XK2)
IF (XMINI.GT.XMK)
F
T
GO TO 80
8 B2
XNK=AMIN1 (XK1,XK2)

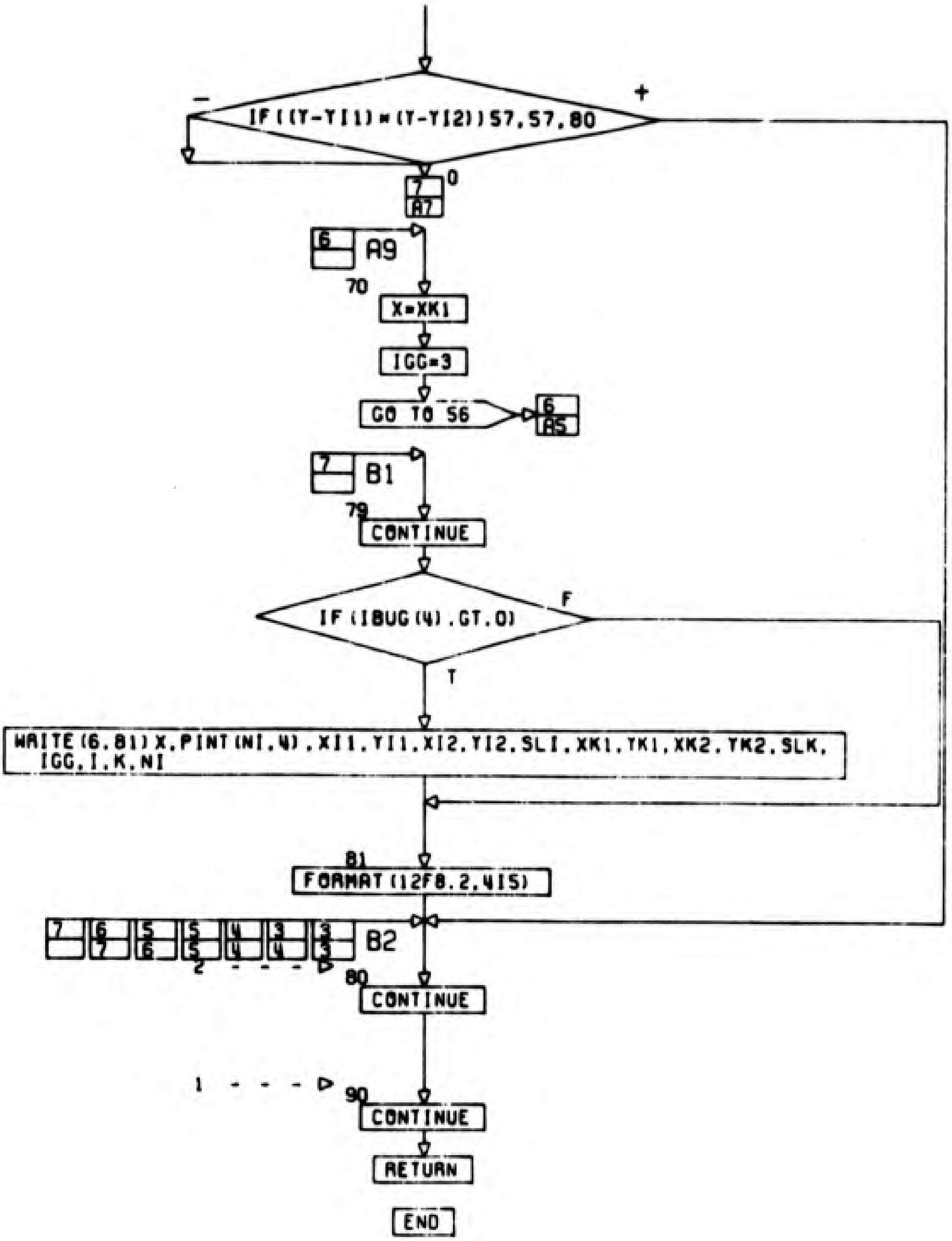

IF (XMAXI.LT.XNK)
F
T
GO TO 80
8
82
YK1=PVRT(LK1,2)
YK2=PVRT(LK2,2)
C CHECK RANGE OF Y
YMK=AMAX1(YK1,YK2)
IF (YMINI.GT.YMK)
T
GO TO 80
8
82
YNK=AMIN1(YK1,YK2)
IF (YMAXI.LT.YNK)
F
T
GO TO 80
8
82
C CHECK FOR COMMON END POINT
ADX=ABS(XMK-XMAXI)-.00001
ADY=ABS(YMK-YMAXI)-.00001


IF (ADX.LE.0..AND.ADY.LE.0.)
F
T
GO TO 80
8
B2
ADX=ABS(XNK-XMINI)-.00001
ADY=ABS(YNK-YMINI)-.00001
IF (ADX.LE.0..AND.ADY.LE.0.)
F
T
GO TO 80
8
B2
C FIND INTERSECTION
DXK=XK2-XK1
ADXI=ABS(DXI)-.0001
ADXK=ABS(DXK)-.0001
IF (ADXI.LE.0.)
F
6
A4
T
IF (ADXK) 80,80,65
−
+
7
A8
8
B2



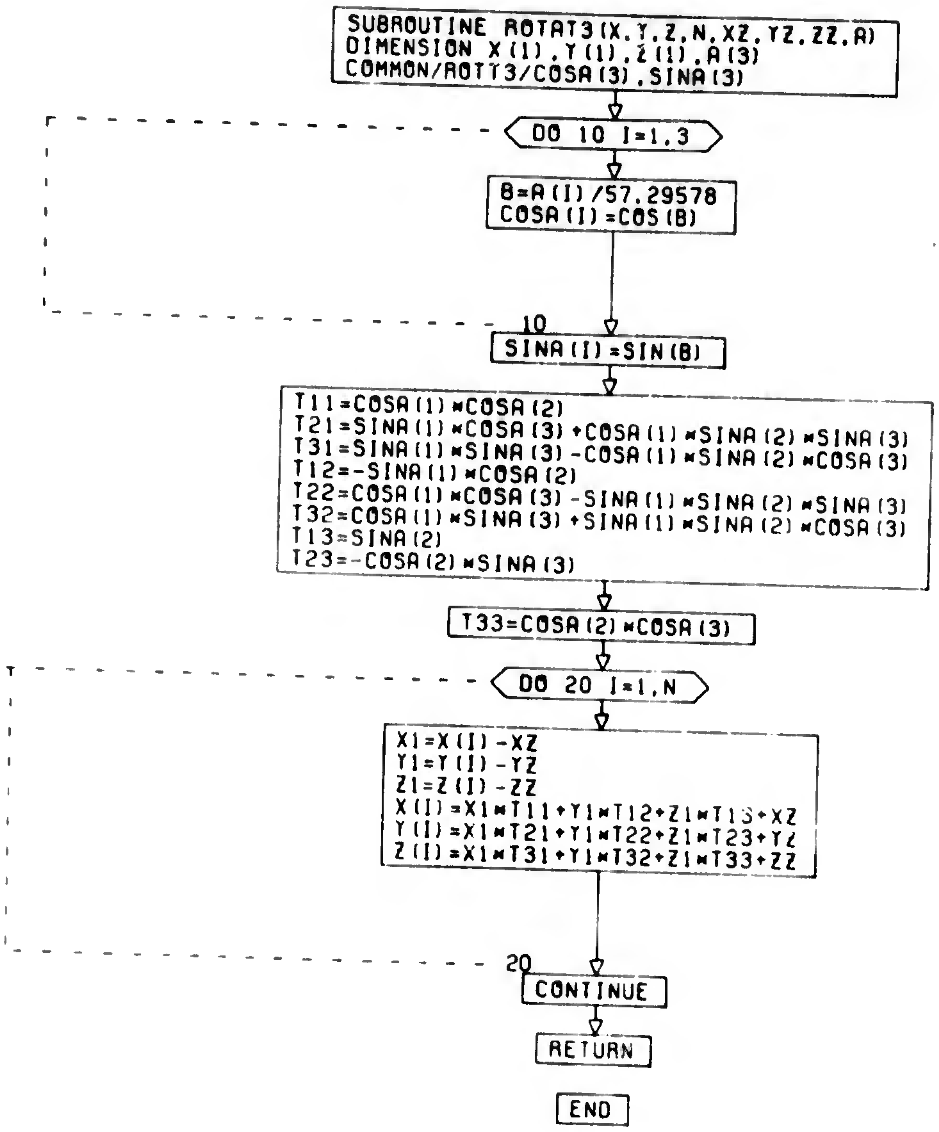
5
A4
IF (ADXK.LE.0.)
F
T
GO TO 70
8
A9
SLK= (YK2-YK1)/DXK
IF (SLI.EQ.SLK)
F
T
GO TO 80
8
B2
C IGG=1 - NEITHER VERTICAL
C 2 - I LINE IS VERTICAL
C 3 - K LINE IS VERTICAL
IGG=1
C X
X= (YI1-YK1-SLI*XI1+SLK*XK1)/(SLK-SLI)
C IS X ON BOTH LINES
A5
8
56
IF ((X-XI1)*(X-XI2).GT.0.)
F
7
A6
T
GO TO 80
8
B2


6
A6
Y=SLI*(X-XI1)+YI1
IF(IGG.EQ.3)
F
T
-
+
IF((Y-YK1)*(Y-YK2))58,80,80
0
8
B2
A7
8
57
IF((X-XK1)*(X-XK2).GT.0.)
F
T
GO TO 80
8
B2
C YES - SAVE POINT
58
NI=NI+1
PINT(NI,1)=I
PINT(NI,2)=K
PINT(NI,3)=X
PINT(NI,4)=Y
NMBI(I)=NMBI(I)+1
NMBI(K)=NMBI(K)+1
GO TO 79
8
B1
5
A8
65
X=XI1
IGG=2
SLK=(YK2-YK1)/DXK
Y=SLK*(X-XK1)+YK1

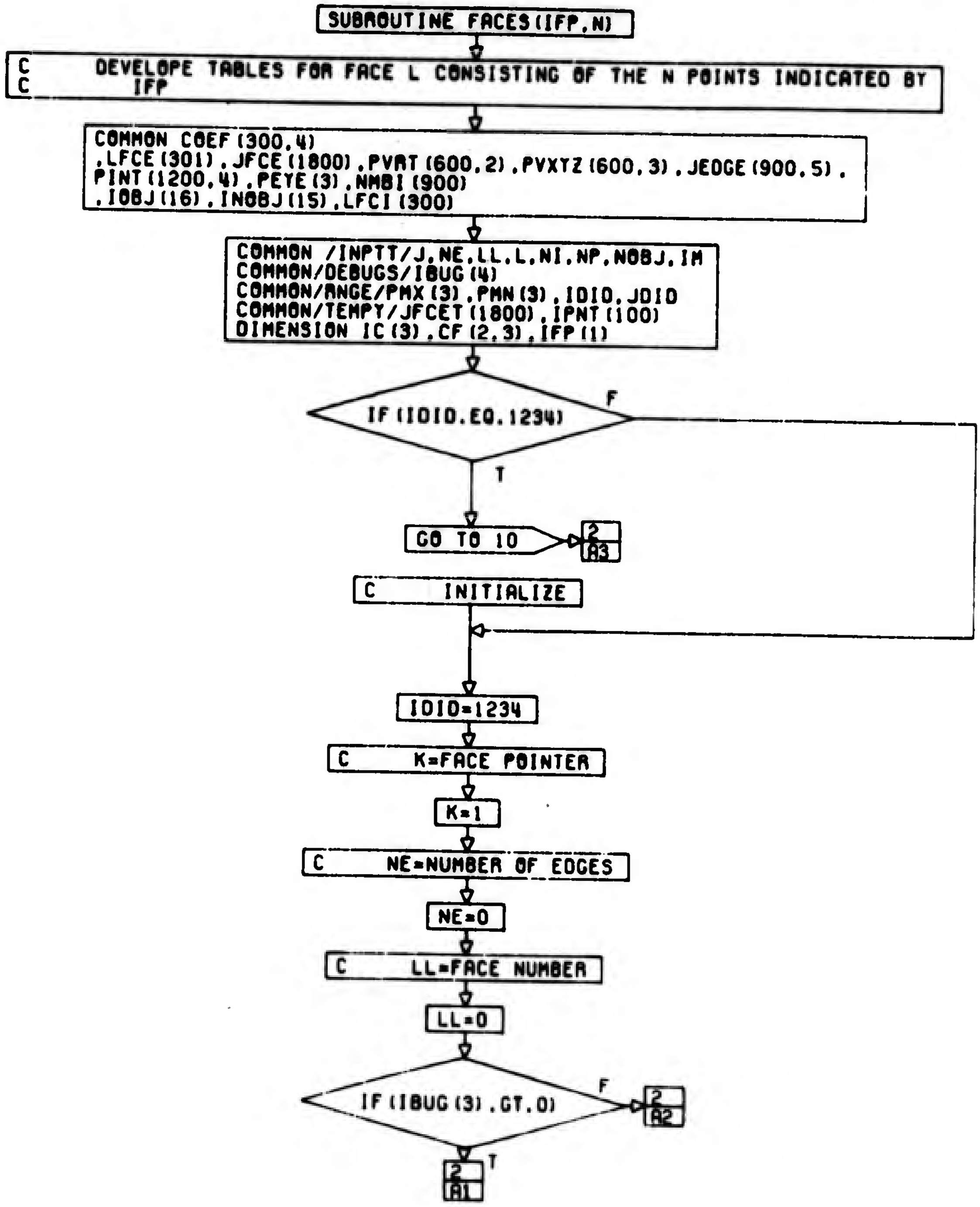


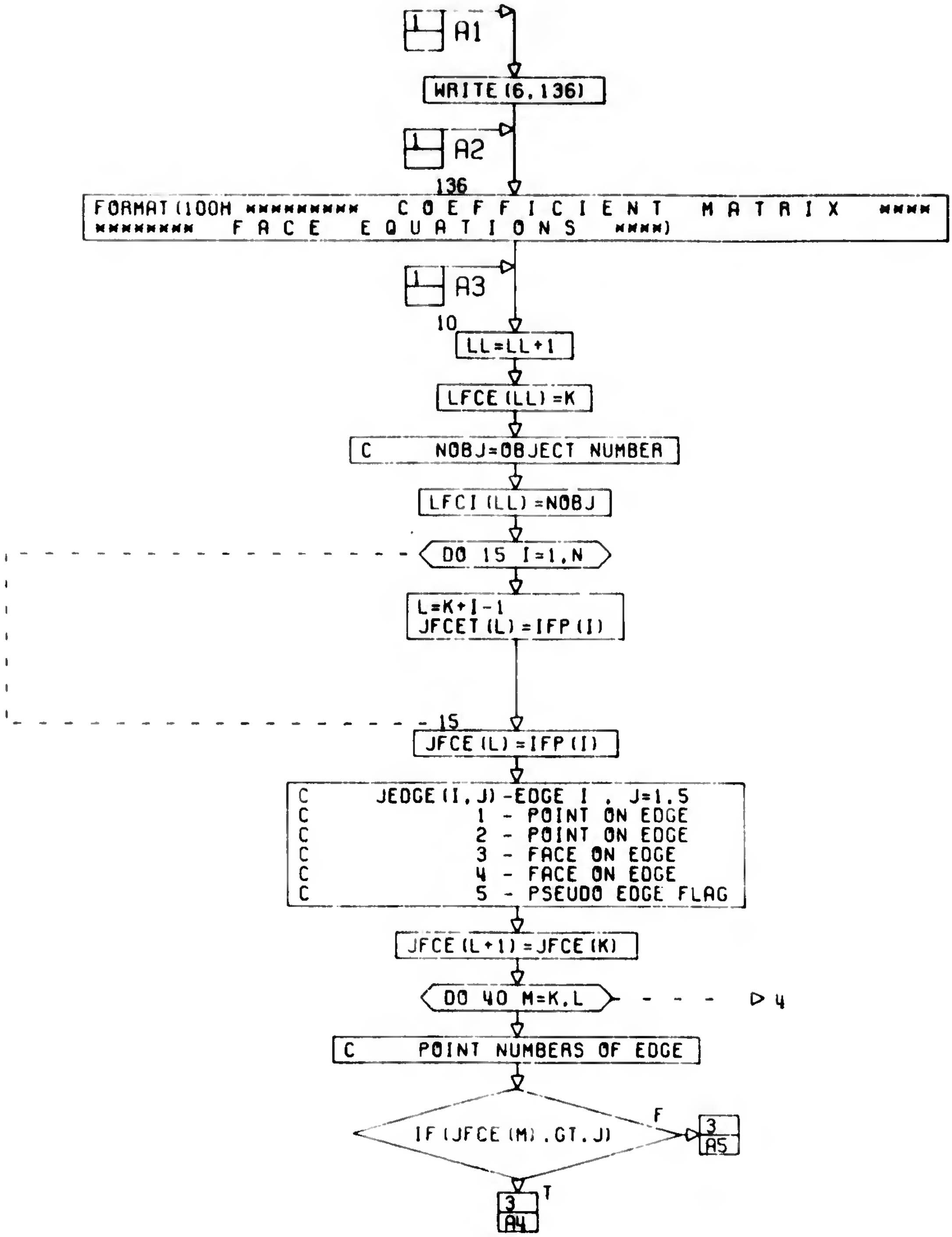
IF((Y-YI1)*(Y-YI2))57,57,80
-
+
0
7
A7
6
A9
70
X=XK1
IGG=3
GO TO 56
6
A5
7
B1
79
CONTINUE
IF(IBUG(4).GT.0)
F
T
WRITE(6,81)X,PINT(NI,4),XI1,YI1,XI2,YI2,SLI,XK1,YK1,XK2,YK2,SLK,
IGG,I,K,NI
81
FORMAT(12F8.2,4I5)
B2
2
80
CONTINUE
1
90
CONTINUE
RETURN
END

SUBROUTINE ROTAT3

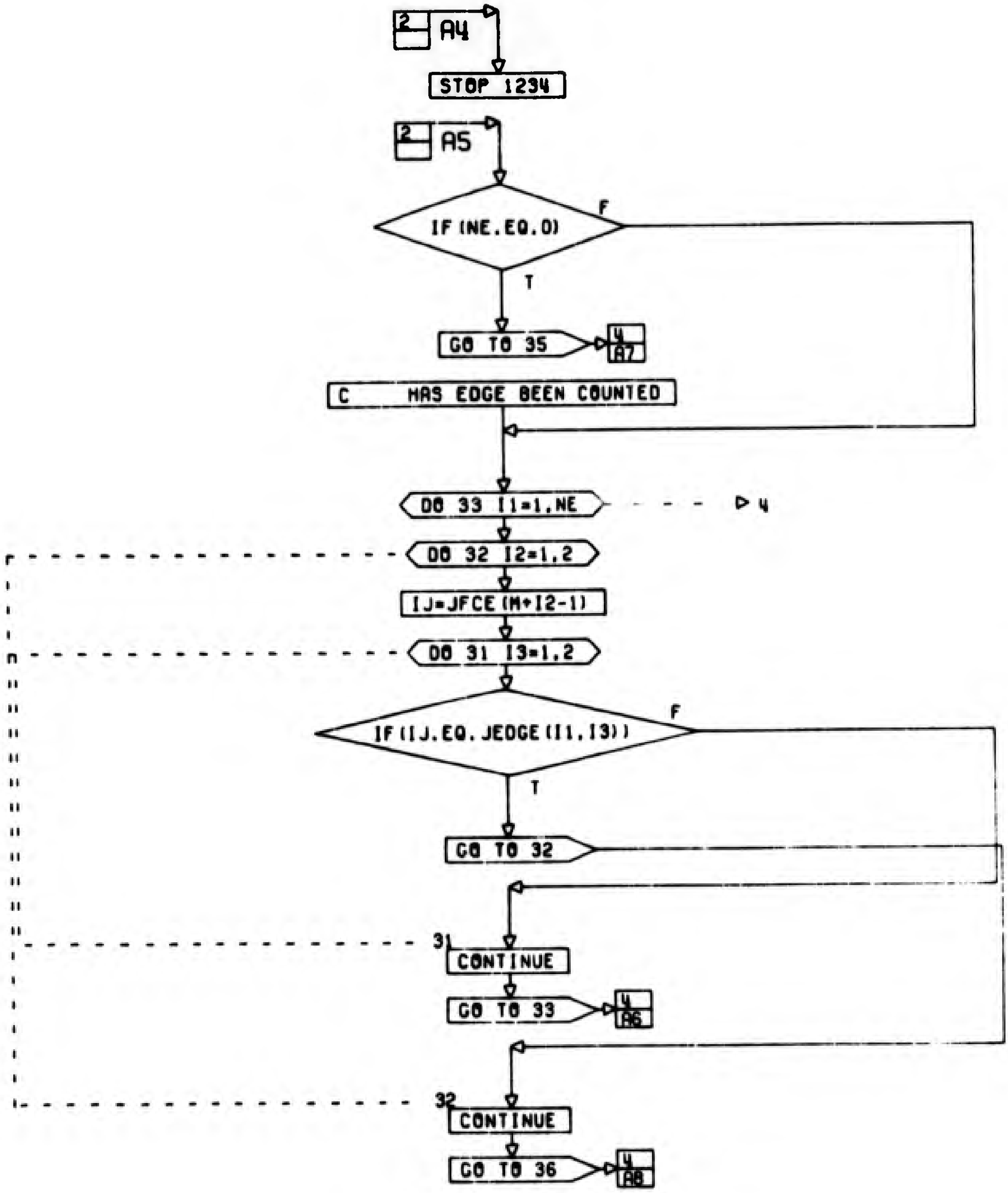


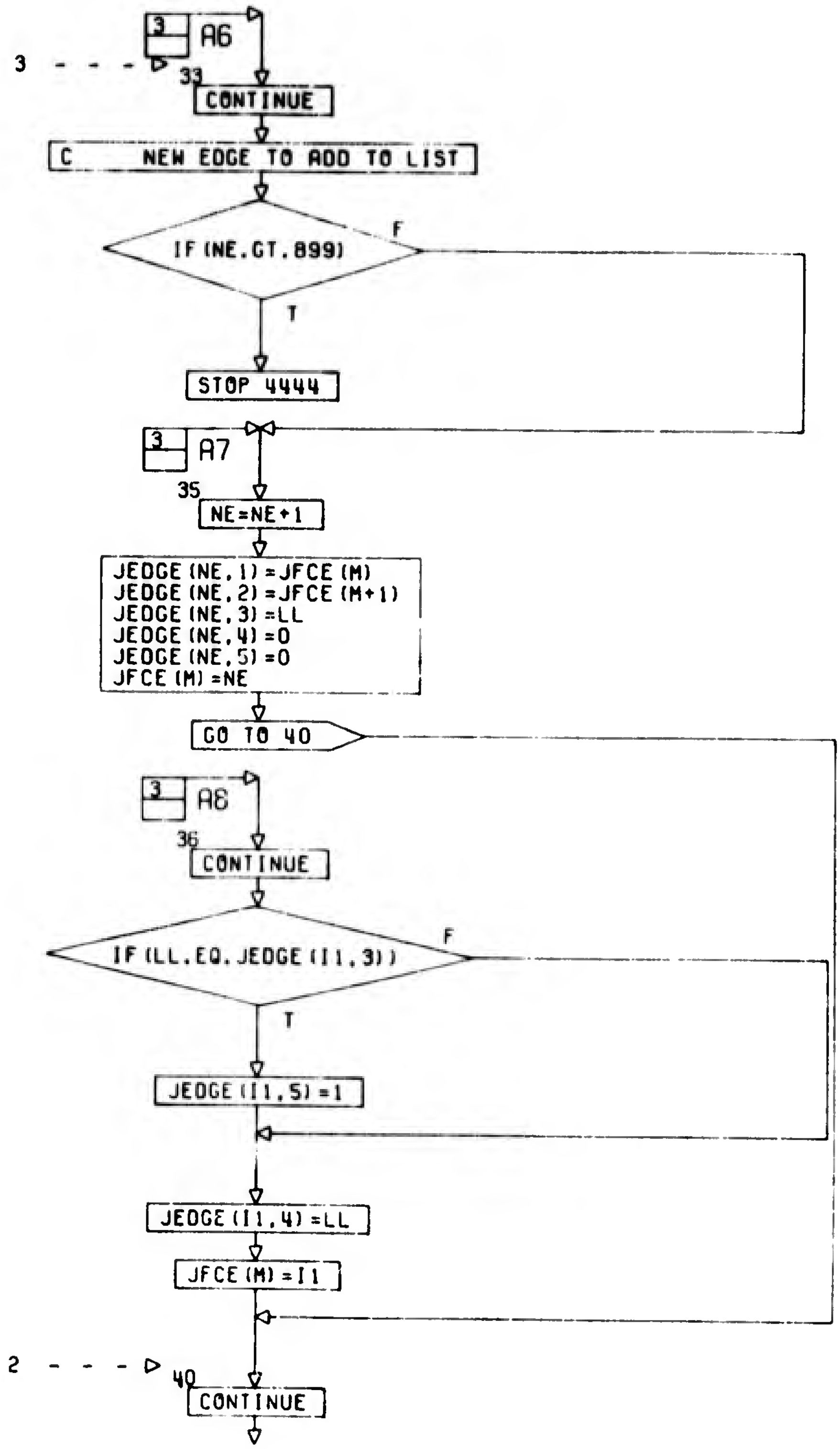
SUBROUTINE FACES

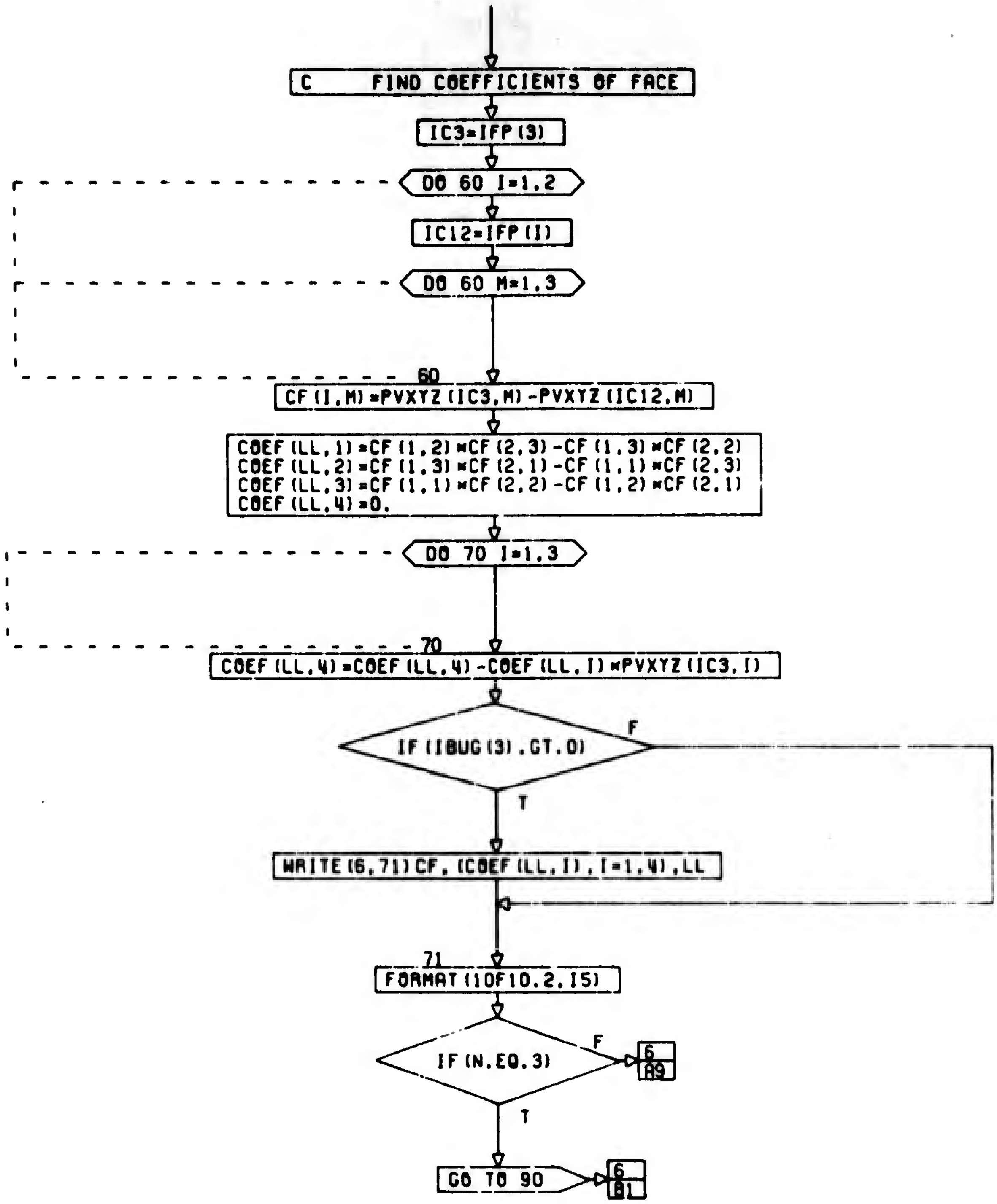




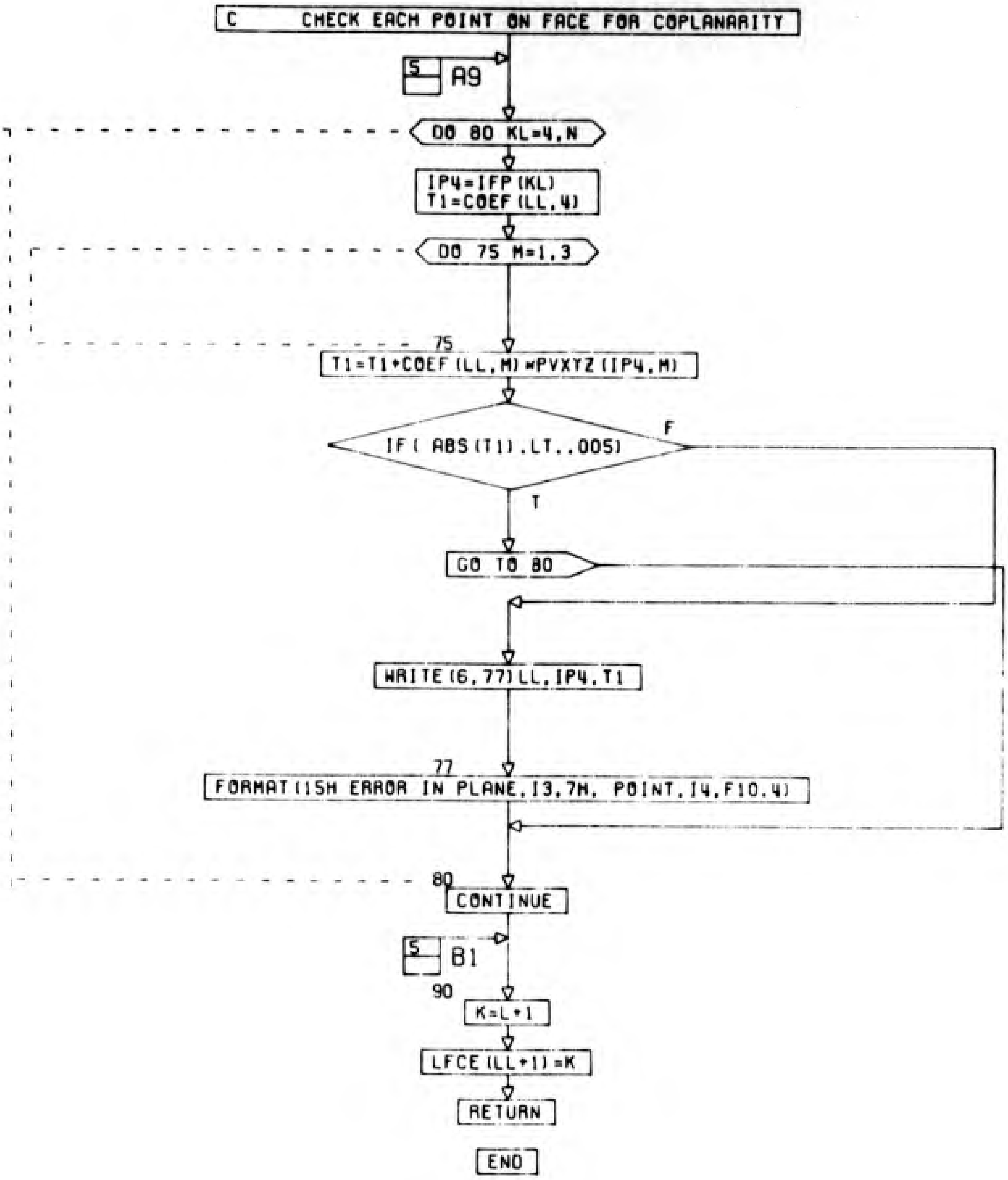
1 A1
WRITE (6,136)
1 A2
136
FORMAT (100H ********* C O E F F I C I E N T M A T R I X **** ******** F A C E E Q U A T I O N S ****)
1 A3
10
LL=LL+1
LFCE (LL) =K
C NOBJ=OBJECT NUMBER
LFCI (LL) =NOBJ
DO 15 I=1,N
L=K+I-1
JFCET (L) =IFP (I)
15
JFCE (L) =IFP (I)
C JEDGE (I,J) -EDGE I , J=1,5
C 1 - POINT ON EDGE
C 2 - POINT ON EDGE
C 3 - FACE ON EDGE
C 4 - FACE ON EDGE
C 5 - PSEUDO EDGE FLAG
JFCE (L+1) =JFCE (K)
DO 40 M=K,L
4
C POINT NUMBERS OF EDGE
IF (JFCE (M) .GT. J)
F
3 A5
T
3 A4


A4
STOP 1234
A5
IF (NE.EQ.0)
F
T
GO TO 35
4
A7
C HAS EDGE BEEN COUNTED
DO 33 I1=1,NE
4
DO 32 I2=1,2
IJ=JFCE(M+I2-1)
DO 31 I3=1,2
IF (IJ.EQ.JEDGE(I1,I3))
F
T
GO TO 32
31
CONTINUE
GO TO 33
4
A6
32
CONTINUE
GO TO 36
4
A8

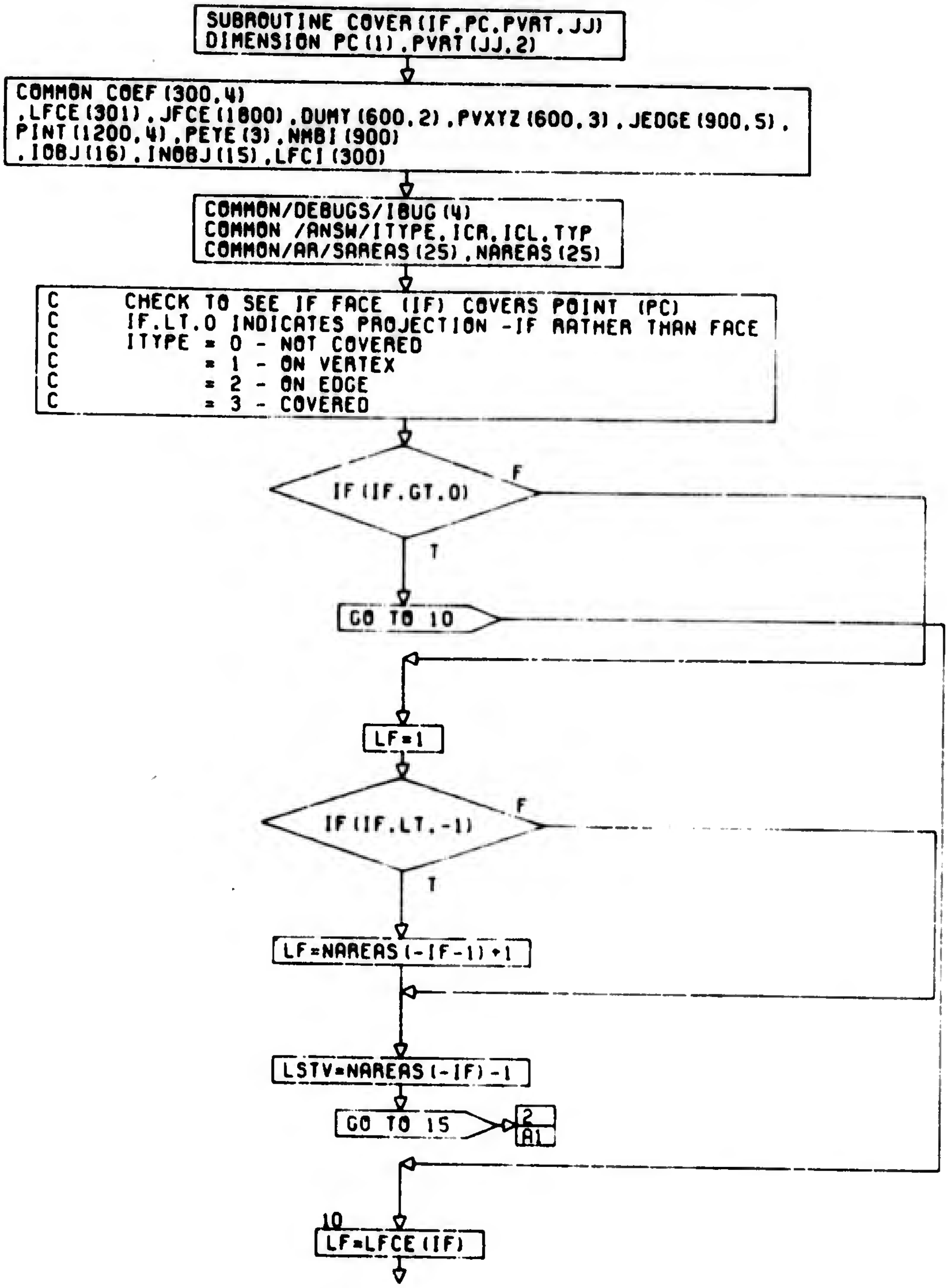

3 A6
3
33
CONTINUE
C NEW EDGE TO ADD TO LIST
IF (NE.GT.899)
F
T
STOP 4444
3 A7
35
NE=NE+1
JEDGE (NE,1)=JFCE(M)
JEDGE (NE,2)=JFCE(M+1)
JEDGE (NE,3)=LL
JEDGE (NE,4)=0
JEDGE (NE,5)=0
JFCE (M)=NE
GO TO 40
3 A8
36
CONTINUE
IF (LL.EQ.JEDGE(I1,3))
F
T
JEDGE (I1,5)=1
JEDGE (I1,4)=LL
JFCE (M)=I1
2
40
CONTINUE

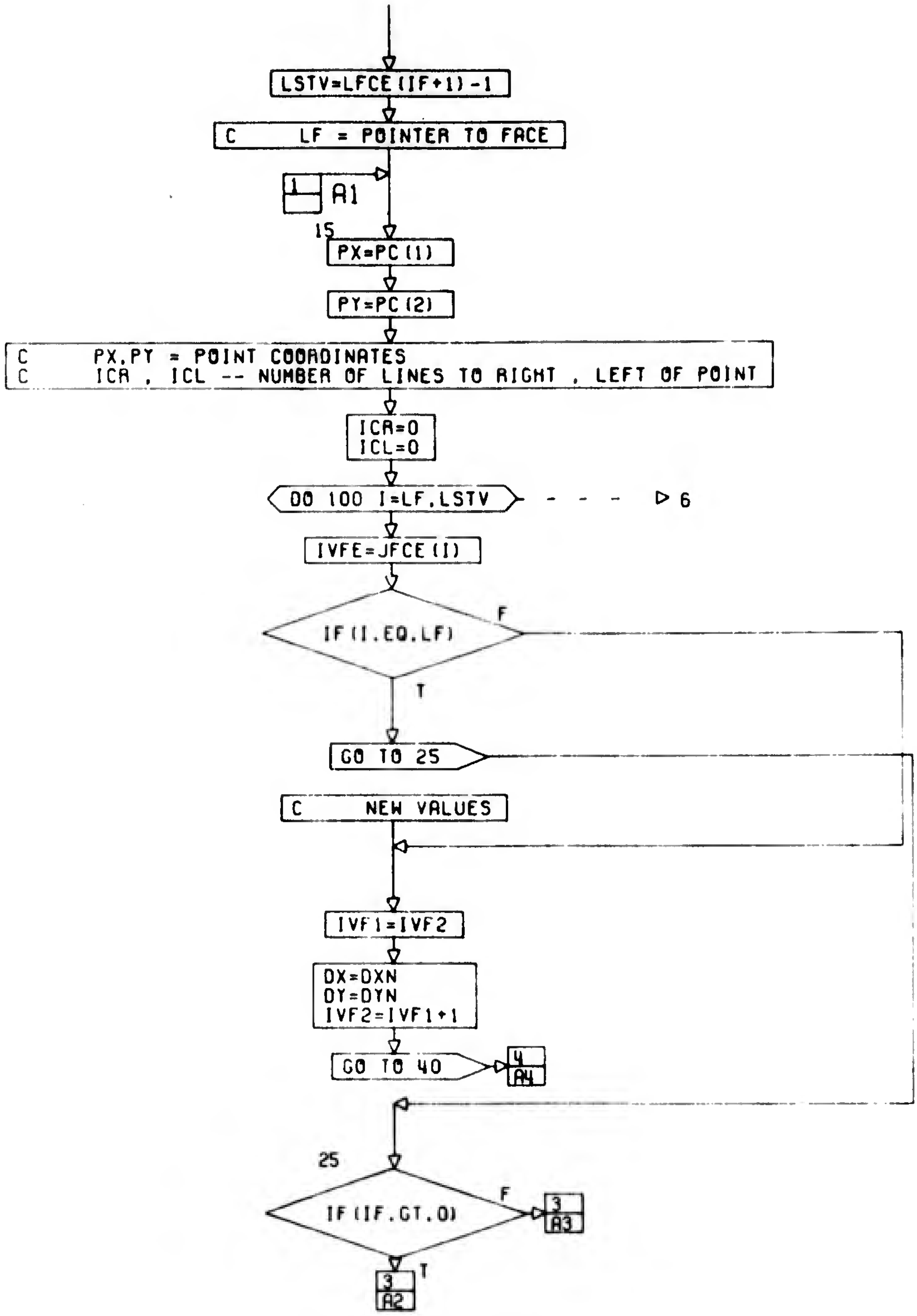


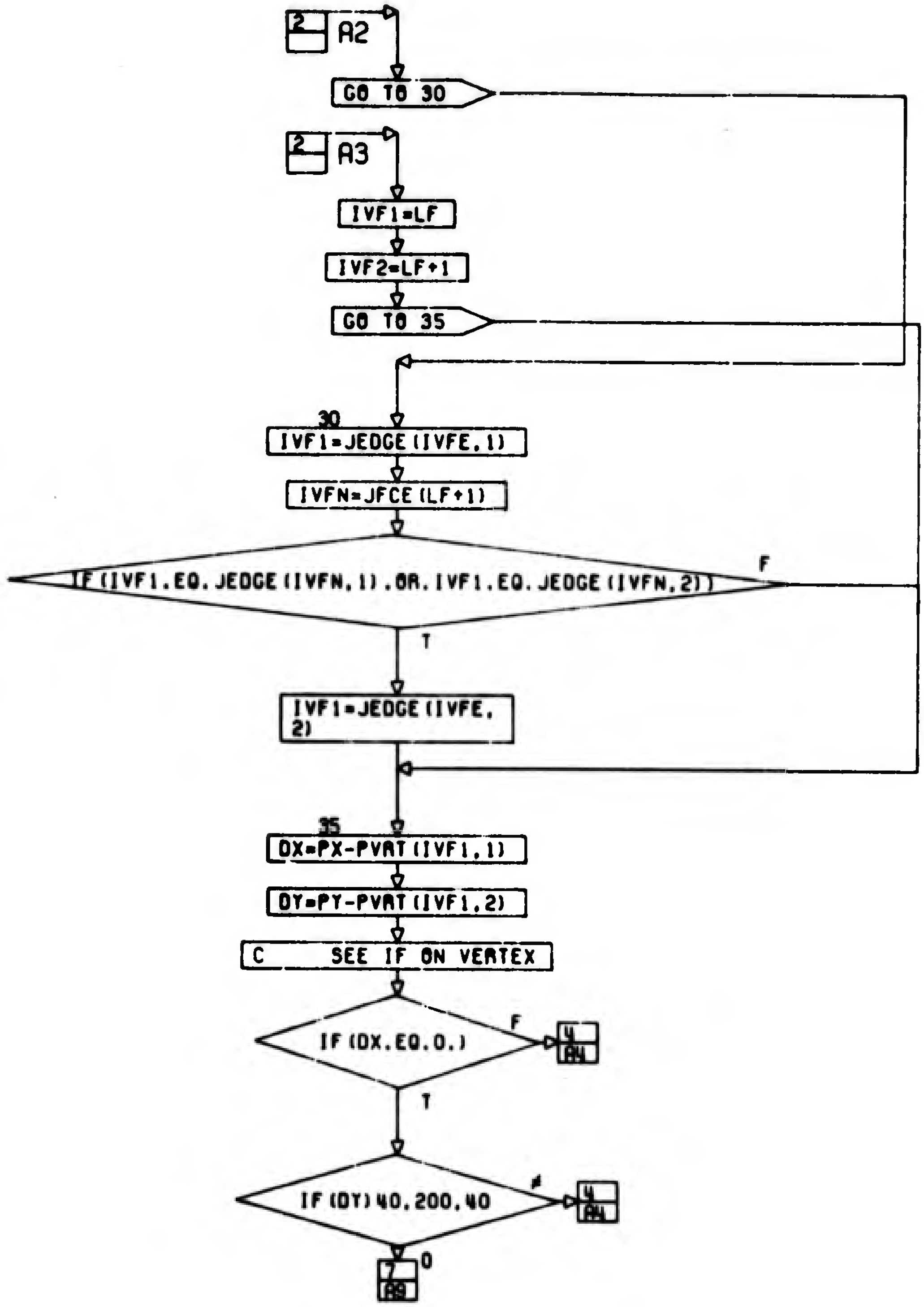
C FIND COEFFICIENTS OF FACE
IC3=IFP(3)
DO 60 I=1,2
IC12=IFP(I)
DO 60 M=1,3
60
CF(I,M)=PVXYZ(IC3,M)-PVXYZ(IC12,M)
COEF(LL,1)=CF(1,2)*CF(2,3)-CF(1,3)*CF(2,2)
COEF(LL,2)=CF(1,3)*CF(2,1)-CF(1,1)*CF(2,3)
COEF(LL,3)=CF(1,1)*CF(2,2)-CF(1,2)*CF(2,1)
COEF(LL,4)=0.
DO 70 I=1,3
70
COEF(LL,4)=COEF(LL,4)-COEF(LL,I)*PVXYZ(IC3,I)
IF(IBUG(3).GT.0)
F
T
WRITE(6,71) CF,(COEF(LL,I),I=1,4),LL
71
FORMAT(10F10.2,I5)
IF(N.EQ.3)
F
6
A9
T
GO TO 90
6
B1


C CHECK EACH POINT ON FACE FOR COPLANARITY
5
A9
DO 80 KL=4,N
IP4=IFP(KL)
T1=COEF(LL,4)
DO 75 M=1,3
75
T1=T1+COEF(LL,M)*PVXYZ(IP4,M)
IF(ABS(T1).LT..005)
F
T
GO TO 80
WRITE(6,77)LL,IP4,T1
77
FORMAT(15H ERROR IN PLANE,I3,7H, POINT,I4,F10.4)
80
CONTINUE
5
B1
90
K=L+1
LFCE(LL+1)=K
RETURN
END

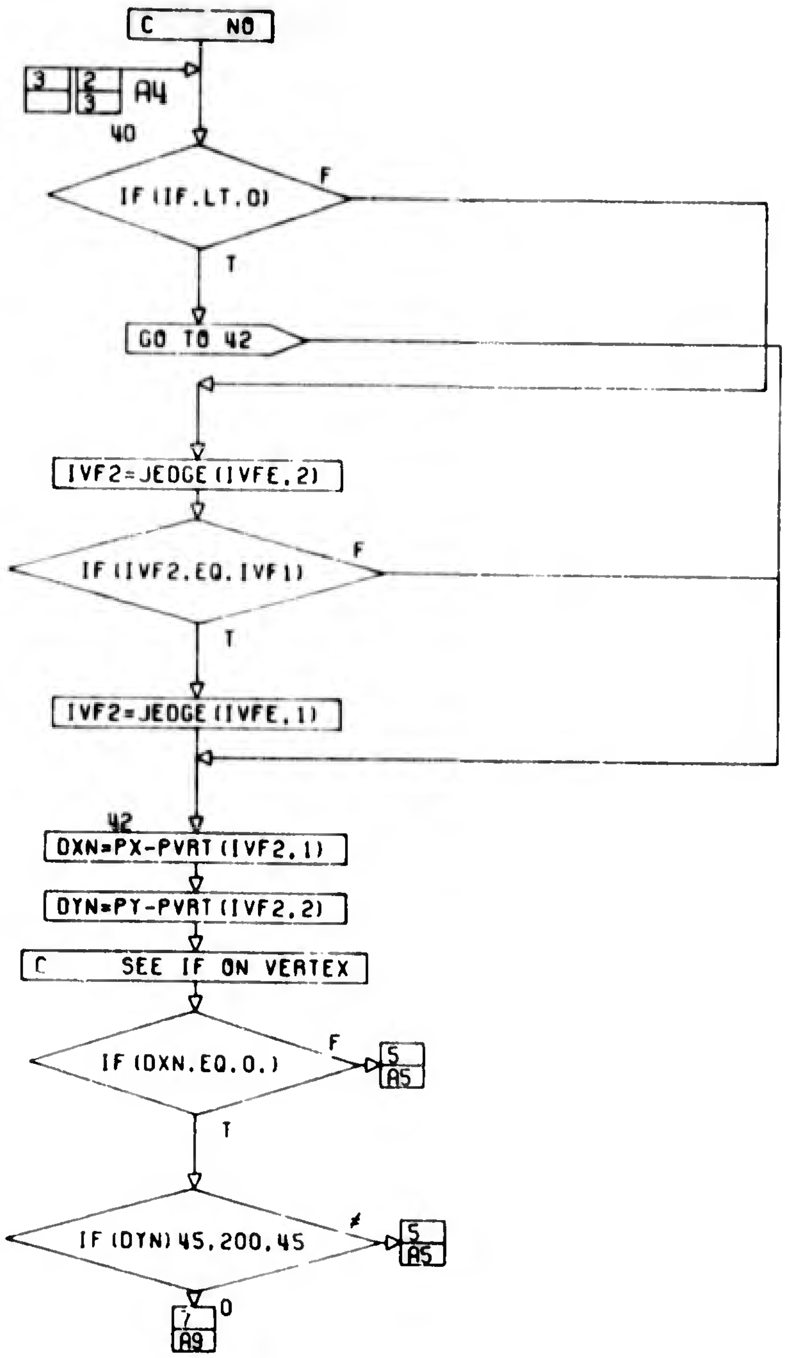
SUBROUTINE COVER

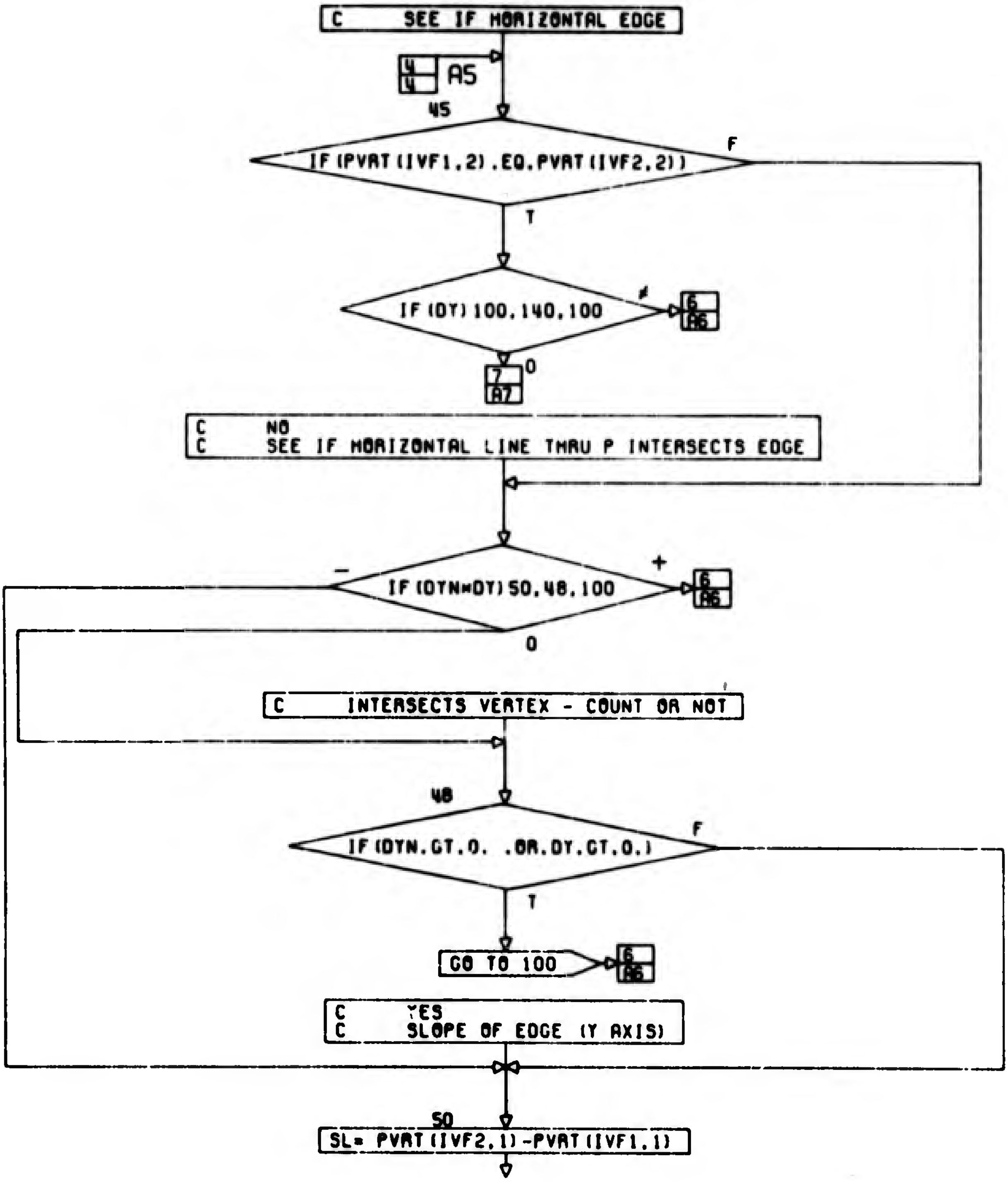



LSTV=LFCE(IF+1)-1
C LF = POINTER TO FACE
1
A1
15
PX=PC(1)
PY=PC(2)
C PX,PY = POINT COORDINATES
C ICR , ICL -- NUMBER OF LINES TO RIGHT , LEFT OF POINT
ICR=0
ICL=0
DO 100 I=LF,LSTV
6
IVFE=JFCE(I)
IF(I.EQ.LF)
F
T
GO TO 25
C NEW VALUES
IVF1=IVF2
DX=DXN
DY=DYN
IVF2=IVF1+1
GO TO 40
4
A4
25
IF(IF.GT.0)
F
3
A3
T
3
A2

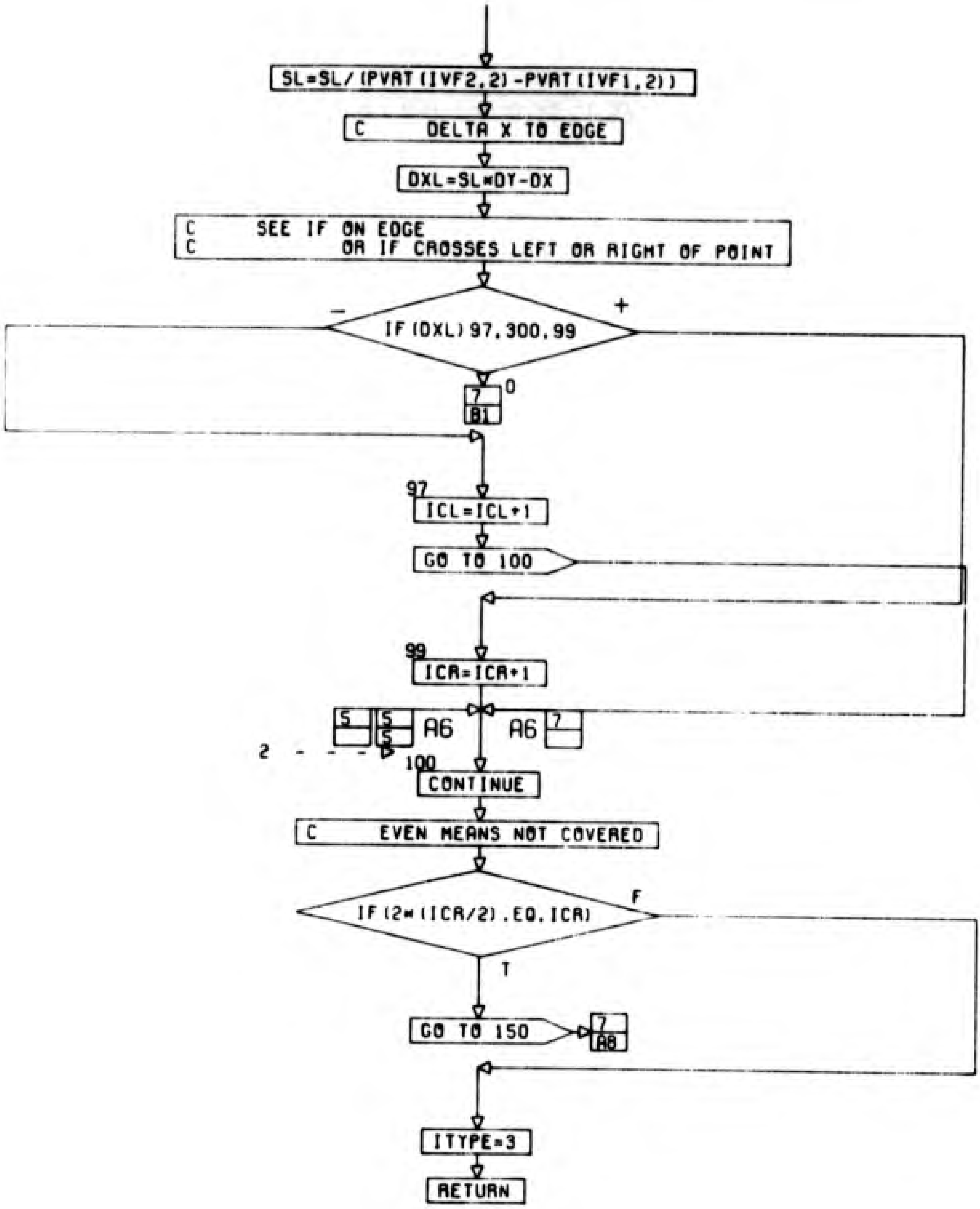


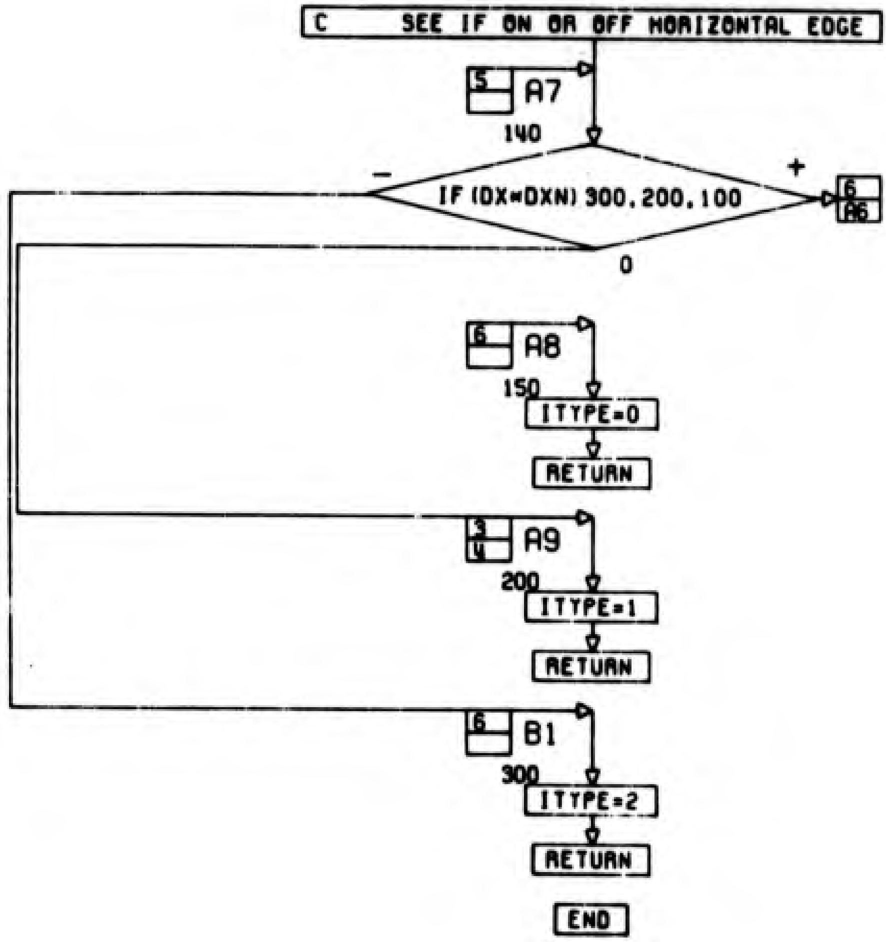
2
A2
GO TO 30
2
A3
IVF1=LF
IVF2=LF+1
GO TO 35
30
IVF1=JEDGE(IVFE,1)
IVFN=JFCE(LF+1)
IF(IVF1.EQ.JEDGE(IVFN,1).OR.IVF1.EQ.JEDGE(IVFN,2))
F
T
IVF1=JEDGE(IVFE,2)
35
DX=PX-PVAT(IVF1,1)
DY=PY-PVAT(IVF1,2)
C SEE IF ON VERTEX
IF(DX.EQ.0.)
F
4
A4
T
IF(DY)40,200,40
≠
4
A4
0
7
A9


C NO
3
2
3
A4
40
IF (IF.LT.0)
F
T
GO TO 42
IVF2=JEDGE(IVFE,2)
IF (IVF2.EQ.IVF1)
F
T
IVF2=JEDGE(IVFE,1)
42
DXN=PX-PVRT(IVF2,1)
DYN=PY-PVRT(IVF2,2)
C SEE IF ON VERTEX
IF (DXN.EQ.0.)
F
5
A5
T
IF (DYN) 45,200,45
≠
5
A5
0
7
A9



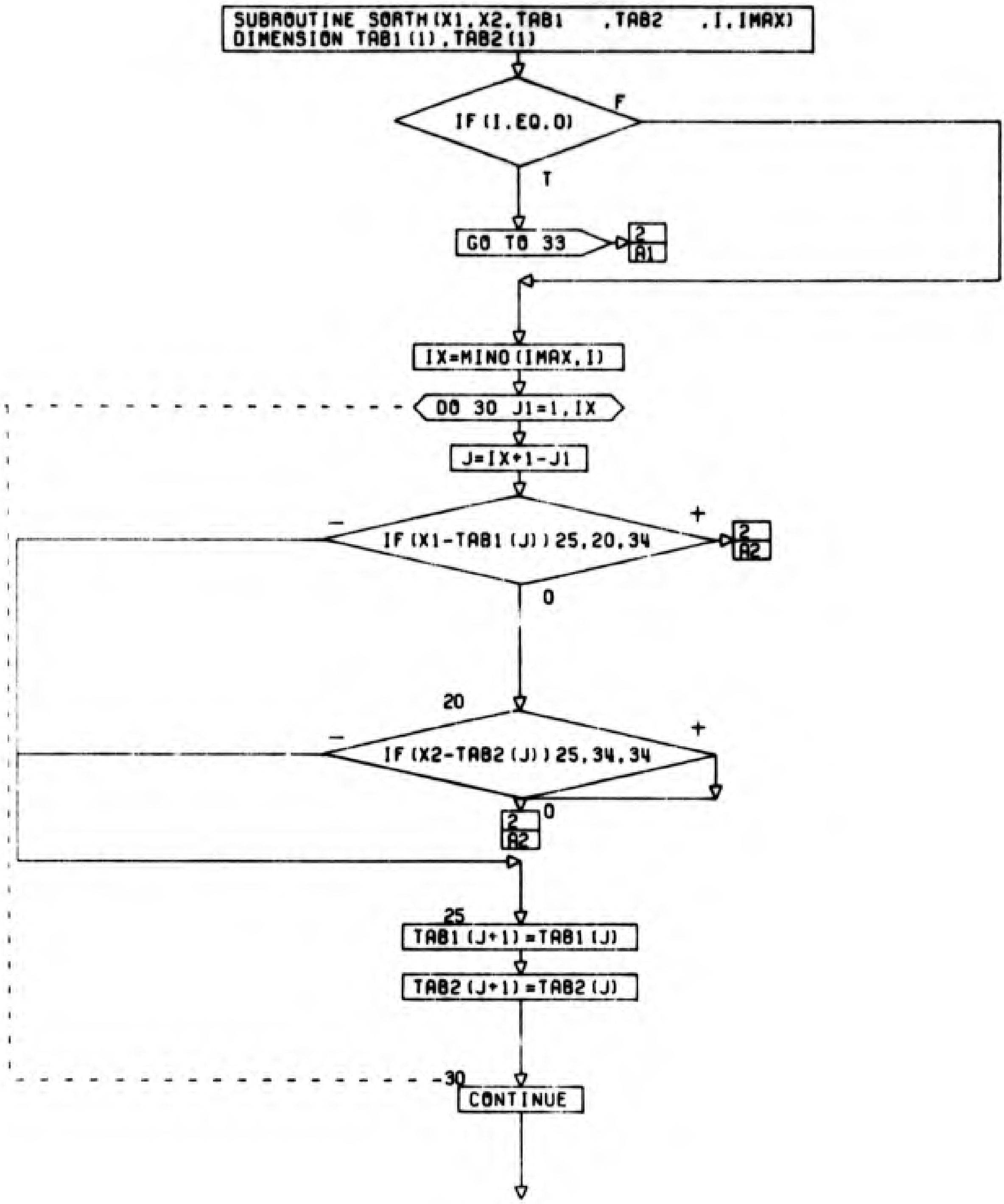
C SEE IF HORIZONTAL EDGE
4
4
A5
45
IF (PVRT(IVF1,2).EQ.PVRT(IVF2,2))
F
T
IF (DY) 100,140,100
≠
6
A6
7
A7
0
C NO
C SEE IF HORIZONTAL LINE THRU P INTERSECTS EDGE
IF (DYN*DY) 50,48,100
−
+
6
A6
0
C INTERSECTS VERTEX - COUNT OR NOT
48
IF (DYN.GT.0. .OR.DY.GT.0.)
F
T
GO TO 100
6
A6
C YES
C SLOPE OF EDGE (Y AXIS)
50
SL= PVRT(IVF2,1)-PVRT(IVF1,1)

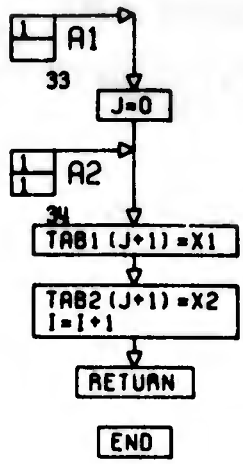

SL=SL/(PVRT(IVF2,2)-PVRT(IVF1,2))
C DELTA X TO EDGE
DXL=SL*DY-DX
C SEE IF ON EDGE
C OR IF CROSSES LEFT OR RIGHT OF POINT
IF (DXL) 97,300,99
-
+
0
7
B1
97
ICL=ICL+1
GO TO 100
99
ICR=ICR+1
5
5
5
A6
A6
7
2
100
CONTINUE
C EVEN MEANS NOT COVERED
IF (2*(ICR/2).EQ.ICR)
F
T
GO TO 150
7
A8
ITYPE=3
RETURN



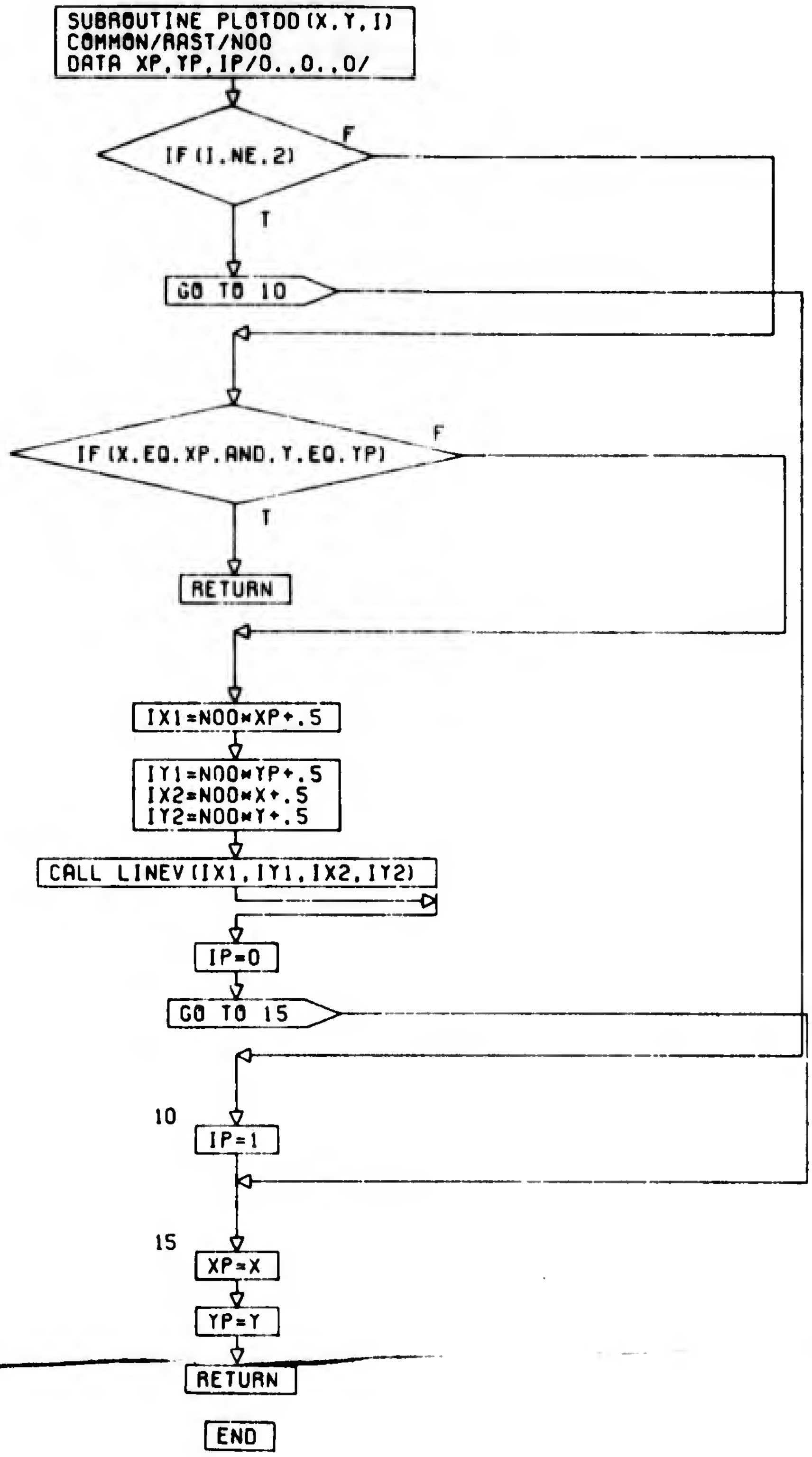
C SEE IF ON OR OFF HORIZONTAL EDGE
5
A7
140
IF (DX-DXN) 300,200,100
-
+
6
A6
0
6
A8
150
ITYPE=0
RETURN
3
4
A9
200
ITYPE=1
RETURN
6
B1
300
ITYPE=2
RETURN
END

SUBROUTINE SORTH

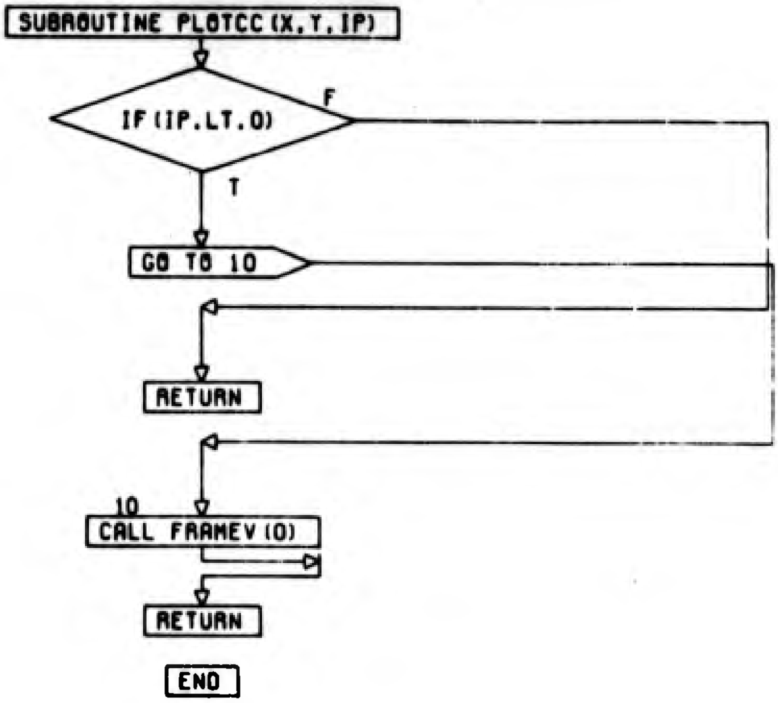



A1
33
J=0
A2
34
TAB1(J+1)=X1
TAB2(J+1)=X2
I=I+1
RETURN
END

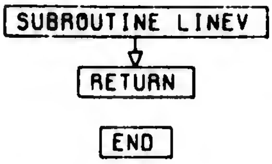
SUBROUTINE PLOTDD



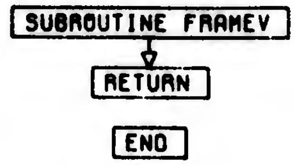
SUBROUTINE PLOTCC



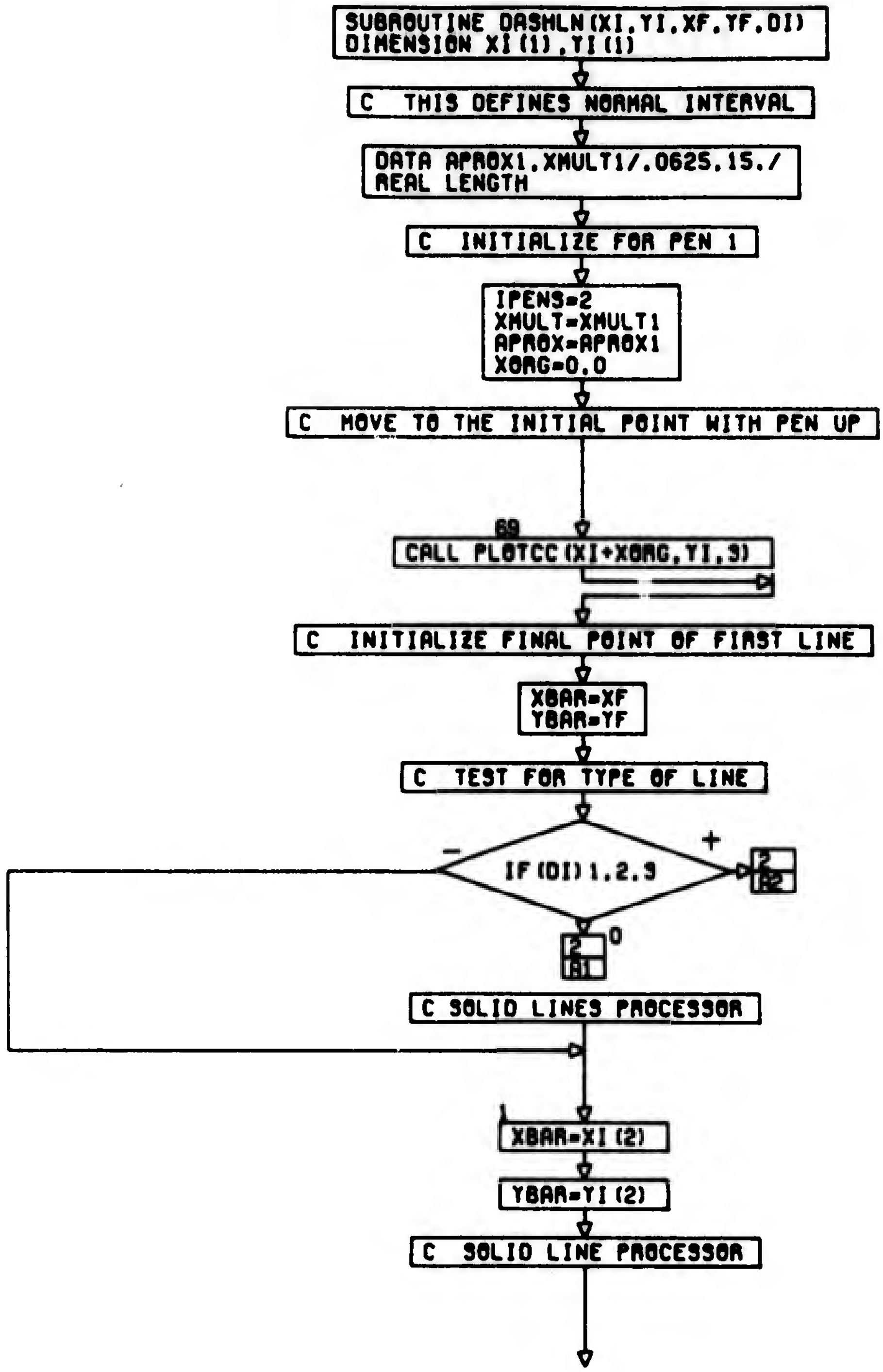
SUBROUTINE LINEV

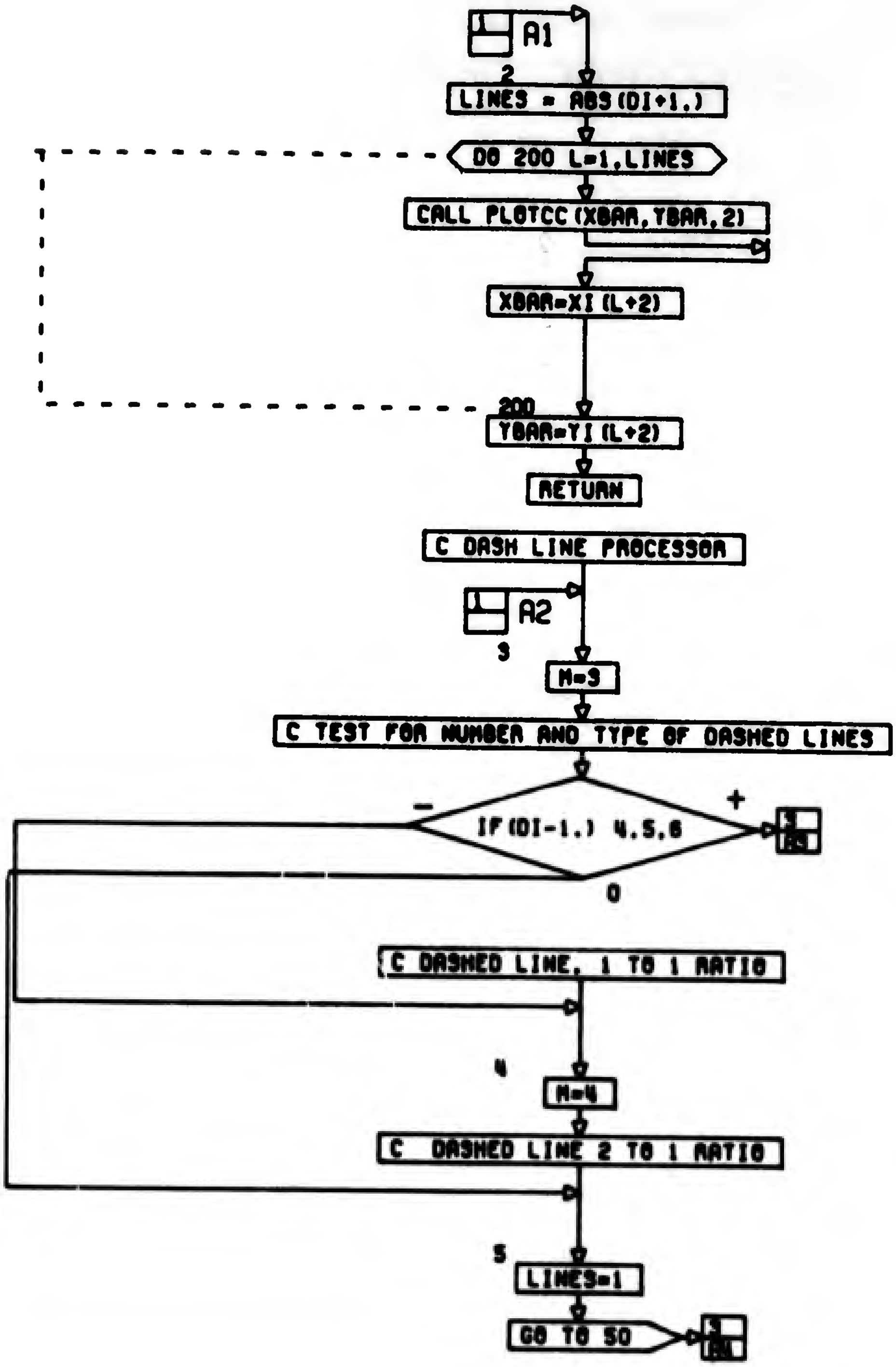


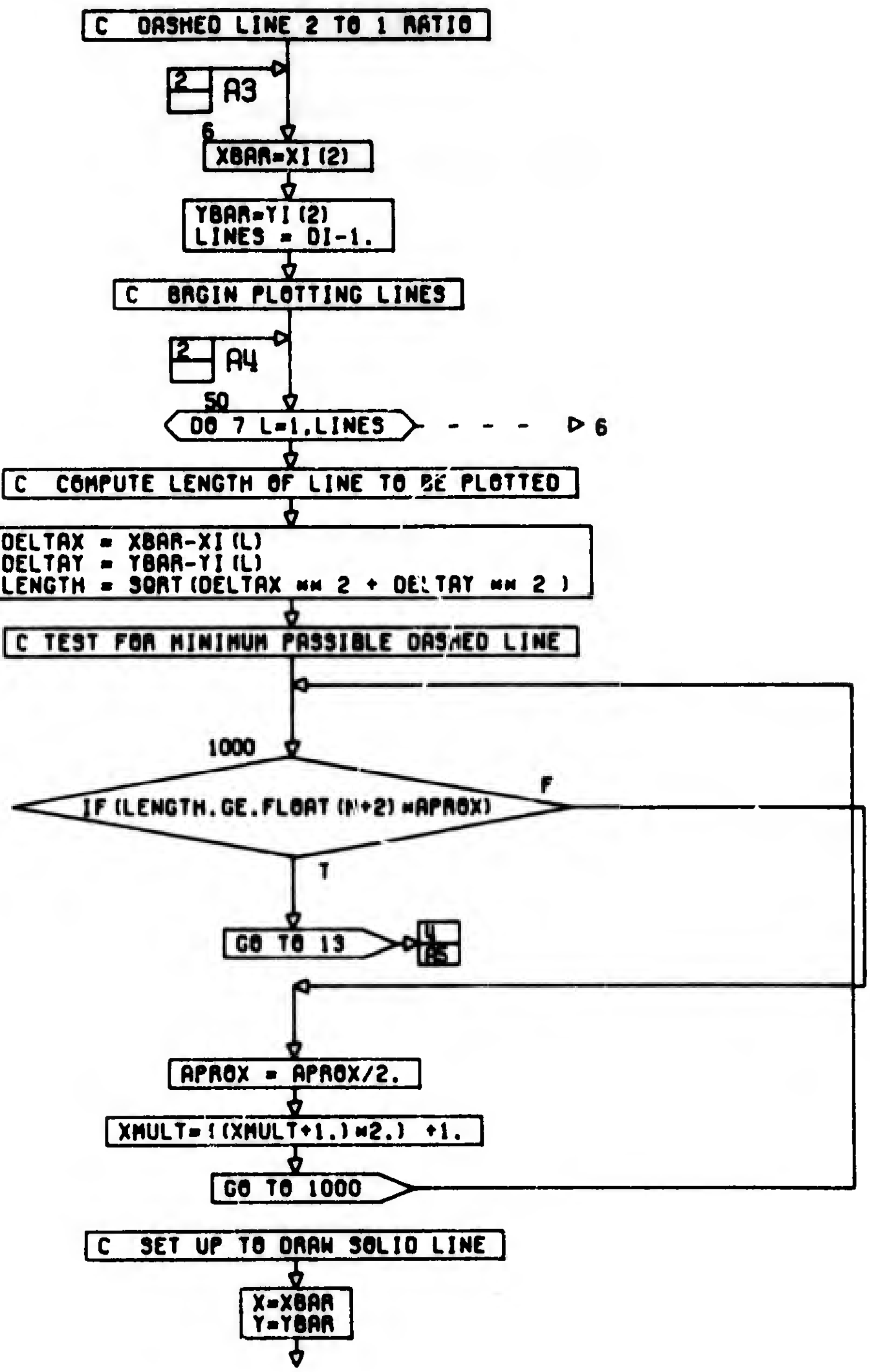
SUBROUTINE FRAMEV

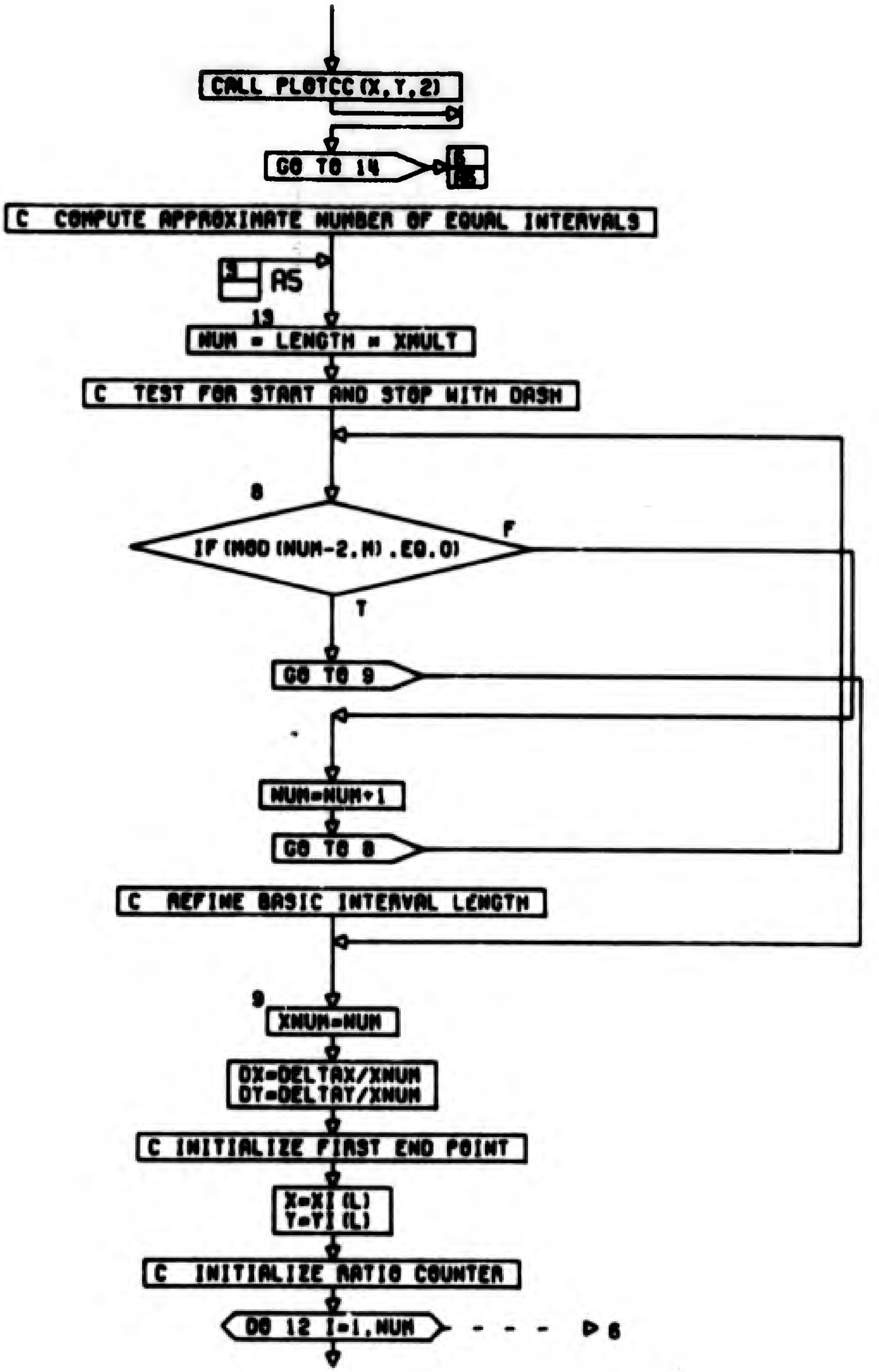


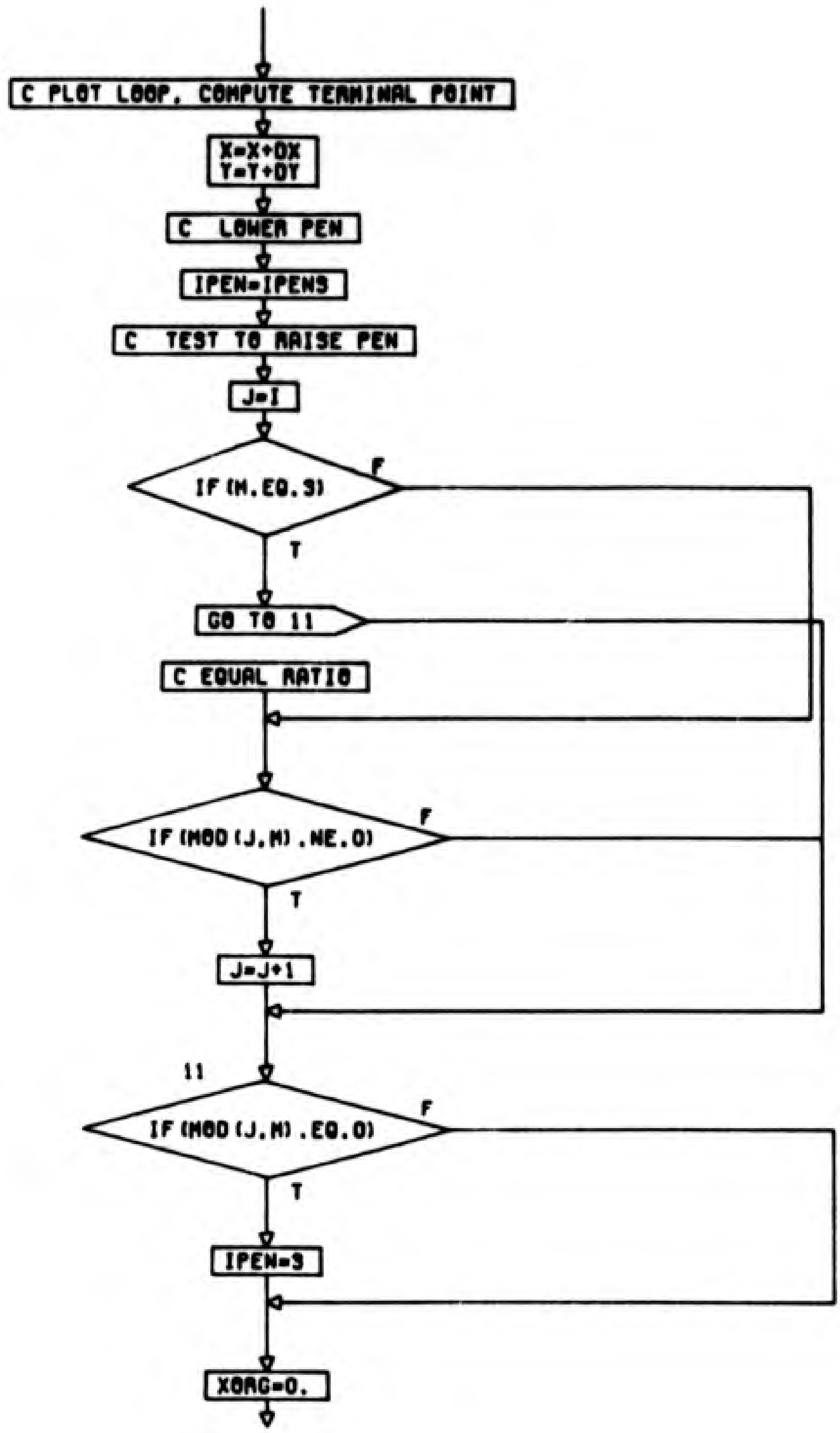
SUBROUTINE DASHLN

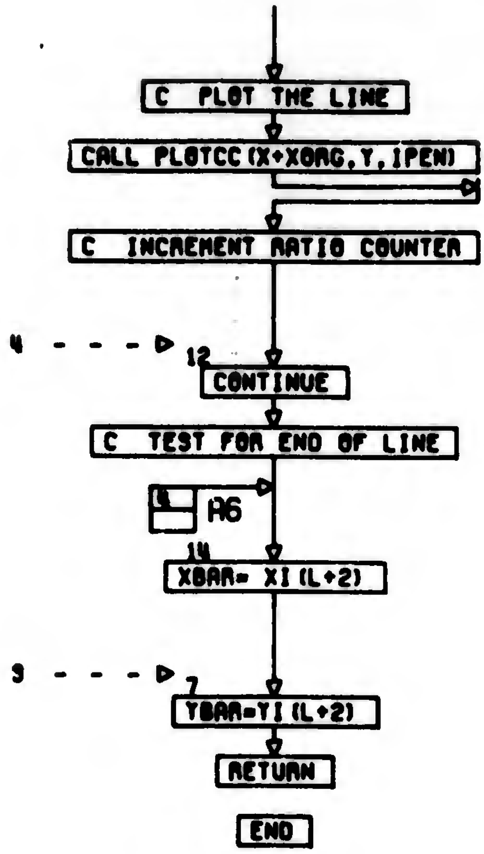



A1
2
LINES = ABS(DI+1.)
DO 200 L=1,LINES
CALL PLOTCC(XBAR,YBAR,2)
XBAR=XI(L+2)
200
YBAR=YI(L+2)
RETURN
C DASH LINE PROCESSOR
A2
3
M=3
C TEST FOR NUMBER AND TYPE OF DASHED LINES
IF(DI-1.) 4,5,6
-
+
0
C DASHED LINE, 1 TO 1 RATIO
4
M=4
C DASHED LINE 2 TO 1 RATIO
5
LINES=1
GO TO 50


C DASHED LINE 2 TO 1 RATIO
2
A3
6
XBAR=XI (2)
YBAR=YI (2)
LINES = DI-1.
C BAGIN PLOTTING LINES
2
A4
50
DO 7 L=1,LINES
6
C COMPUTE LENGTH OF LINE TO BE PLOTTED
DELTAX = XBAR-XI (L)
DELTAY = YBAR-YI (L)
LENGTH = SQRT (DELTAX ** 2 + DELTAY ** 2)
C TEST FOR MINIMUM PASSIBLE DASHED LINE
1000
IF (LENGTH.GE.FLOAT (N+2) *APROX)
F
T
GO TO 13
4
A5
APROX = APROX/2.
XMULT=((XMULT+1.)*2.) +1.
GO TO 1000
C SET UP TO DRAW SOLID LINE
X=XBAR
Y=YBAR

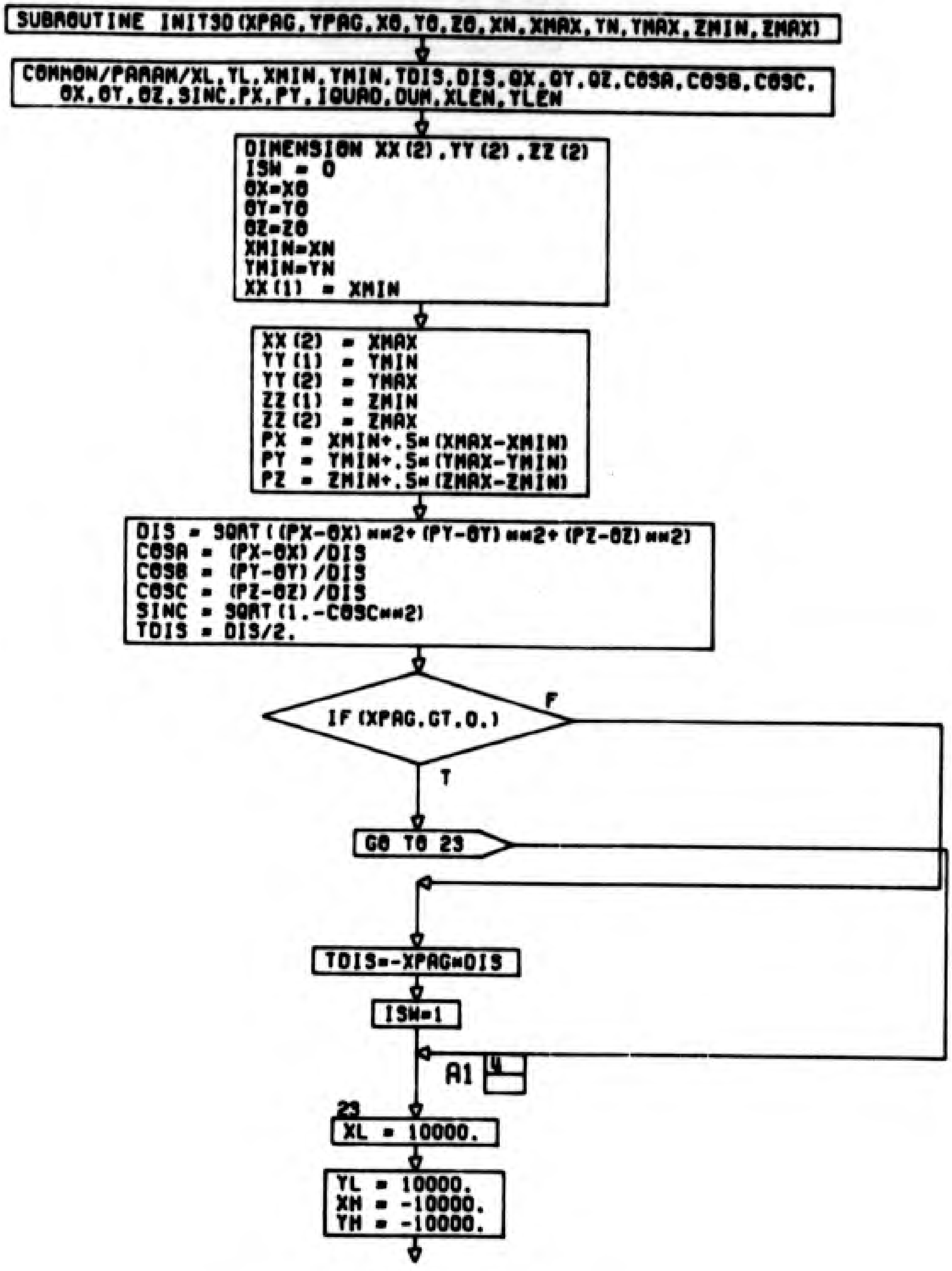

CALL PLOTCC (X,Y,2)
GO TO 14
C COMPUTE APPROXIMATE NUMBER OF EQUAL INTERVALS
A5
13
NUM = LENGTH * XMULT
C TEST FOR START AND STOP WITH DASH
8
IF (MOD (NUM-2,M) .EQ.0)
F
T
GO TO 9
NUM=NUM+1
GO TO 8
C REFINE BASIC INTERVAL LENGTH
9
XNUM=NUM
DX=DELTAX/XNUM
DY=DELTAY/XNUM
C INITIALIZE FIRST END POINT
X=XI (L)
Y=YI (L)
C INITIALIZE RATIO COUNTER
DO 12 I=1,NUM
6

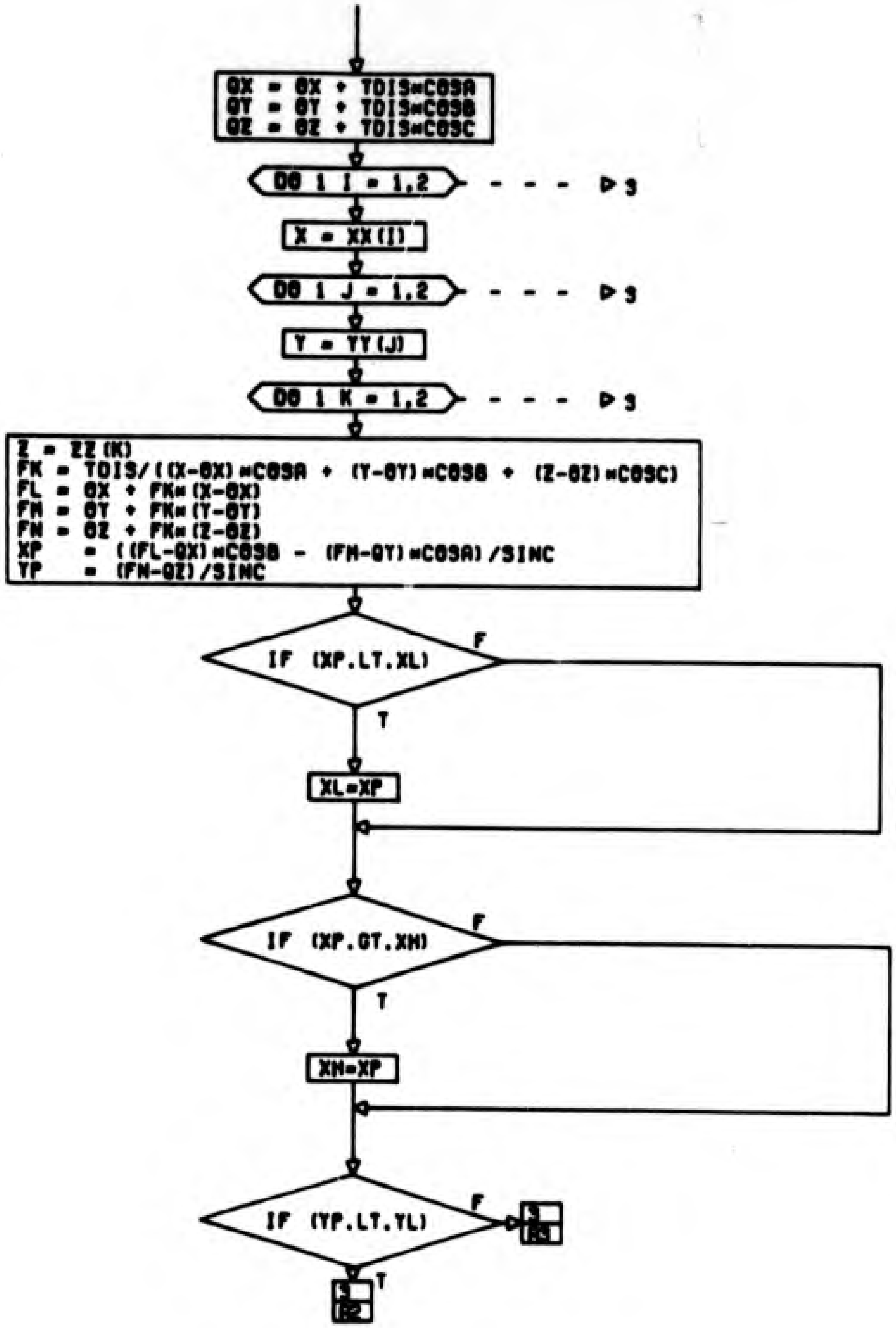

C PLOT LOOP, COMPUTE TERMINAL POINT
X=X+DX
Y=Y+DY
C LOWER PEN
IPEN=IPENS
C TEST TO RAISE PEN
J=I
IF (M.EQ.3)
F
T
GO TO 11
C EQUAL RATIO
IF (MOD (J,M) .NE.0)
F
T
J=J+1
11
IF (MOD (J,M) .EQ.0)
F
T
IPEN=3
XORG=0.

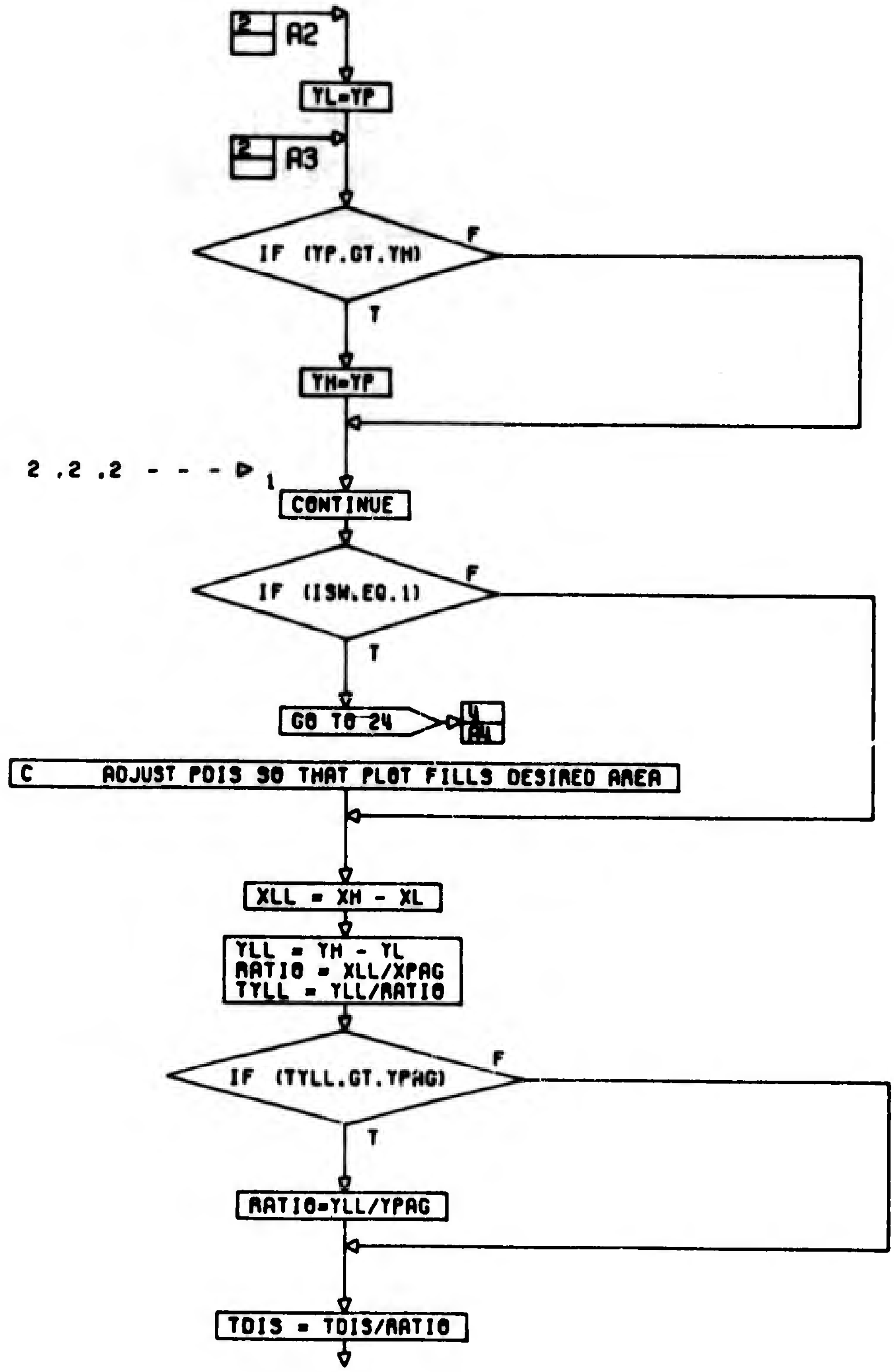


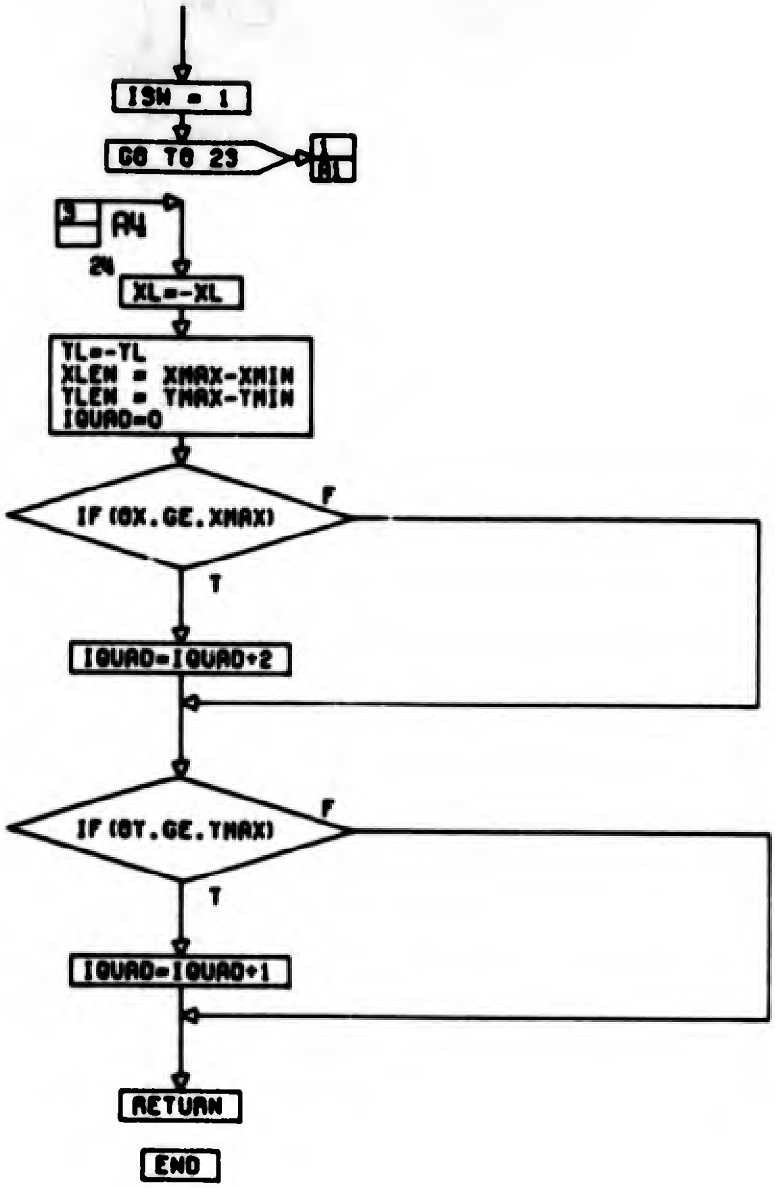
C PLOT THE LINE
CALL PLOTCC(X+XORG,Y,IPEN)
C INCREMENT RATIO COUNTER
4
12
CONTINUE
C TEST FOR END OF LINE
A6
14
XBAR= XI(L+2)
9
7
YBAR=YI(L+2)
RETURN
END

SUBROUTINE INIT3D



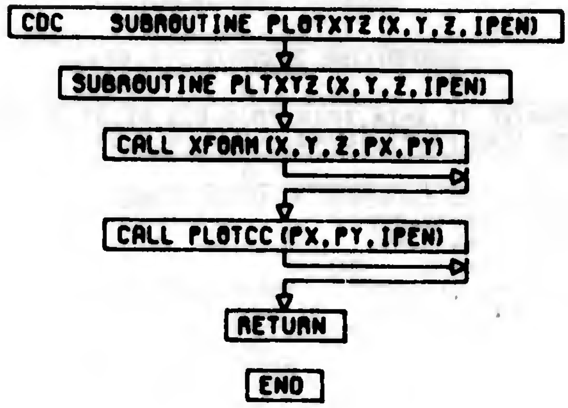

QX = OX + TDIS*COSA
QY = OY + TDIS*COSB
QZ = OZ + TDIS*COSC
DO 1 I = 1,2
3
X = XX(I)
DO 1 J = 1,2
3
Y = YY(J)
DO 1 K = 1,2
3
Z = ZZ(K)
FK = TDIS/((X-OX)*COSA + (Y-OY)*COSB + (Z-OZ)*COSC)
FL = OX + FK*(X-OX)
FM = OY + FK*(Y-OY)
FN = OZ + FK*(Z-OZ)
XP = ((FL-QX)*COSB - (FM-QY)*COSA)/SINC
YP = (FN-QZ)/SINC
IF (XP.LT.XL)
F
T
XL=XP
IF (XP.GT.XH)
F
T
XH=XP
IF (YP.LT.YL)
F
T
3
3


A2
YL=YP
A3
IF (YP.GT.YM)
F
T
YM=YP
2 ,2 ,2
1
CONTINUE
IF (ISW.EQ.1)
F
T
GO TO 24
A4
C ADJUST PDIS SO THAT PLOT FILLS DESIRED AREA
XLL = XM - XL
YLL = YM - YL
RATIO = XLL/XPAG
TYLL = YLL/RATIO
IF (TYLL.GT.YPAG)
F
T
RATIO=YLL/YPAG
TDIS = TDIS/RATIO



ISW = 1
GO TO 23
1
A1
3
A4
24
XL=-XL
YL=-YL
XLEN = XMAX-XMIN
YLEN = YMAX-YMIN
IQUAD=0
IF (OX.GE.XMAX)
F
T
IQUAD=IQUAD+2
IF (OY.GE.YMAX)
F
T
IQUAD=IQUAD+1
RETURN
END

SUBROUTINE PLOTXYZ



SUBROUTINE XFORM

```
SUBROUTINE XFORM(X,Y,Z,DX,DY)

COMMON/PARAM/XL,YL,XMIN,YMIN,TDIS,DIS,OX,OY,OZ,COSA,COSB,COSC,
   QX,QY,QZ,SINC,PX,PY,IQUAD,DUM,XLEN,YLEN

FK = TDIS/((X-OX)*COSA + (Y-OY)*COSB + (Z - OZ)*COSC)
FL = OX + FK*(X-OX)
FM = OY + FK*(Y-OY)
FN = OZ + FK*(Z    -OZ)
DX    = ((FL-QX)*COSB - (FM-QY)*COSA)/SINC + XL
DY    = (FN-QZ)/SINC + YL

RETURN

END
```

APPENDIX III

COMMON AND DIMENSION BLOCKS

/Blank/CØEF (m, 4), LFCE (m + 1), JFCE (3n), PVRT (n,2),

PVXYZ (n,3), JEDGE (3n/2, 5), PINT (2n, 4), PEYE (3),

NMBI (3n/2), IØBJ (k + 1), INØBJ (k), LFCI (m)

where m = maximum number of faces allowed (300)

n = maximum number of points allowed (600)

k = maximum number of objects allowed (15)

CØEF - Coefficients for each face

(,1) - X

(,2) - Y

(,3) - Z

(,4) - Distance of viewing point to plane of face

LFCE - Face pointer to edges

JFCE - Edge numbers for each face

PVRT - Projected coordinates of each point (in inches)

(,1) - X

(,2) - Y

PVXYZ - Three space coordinates of each point (in input units)

(,1) - X

(,2) - Y

(,3) - Z

JEDGE - Pointers for each edge

(,1) - First point number on edge

(,2) - Second point number on edge

(,3) - First face on edge

(,4) - Second face on edge

(,5) - Pseudo edge flag

PINT - Intersection coordinates of the projected edges

(,1) - First edge number on intersection

(,2) - Second edge number on intersection

(,3) - X value of intersection

(,4) - Y value of intersection

PEYE - Three space coordinates of the viewing point

(,1) - X

(,2) - Y

(,3) - Z

NMBI - Number of intersections on each edge

IØBJ - Pointer to first face in object

INØBJ - Type code for each object

LFCI - Object number of each face

/TEMPY/JFCET (3n), IPNT (j)

where n = (see before)

j = maximum number of points on any object (100)

JFCET - point numbers for each face

IPNT - point numbers for each object

/RNGE/PMX(3), PMN(3), IDID, JDID

PMX - maximum input coordinates

(1) - X

(2) - Y

(3) - Z

PMN - minimum input coordinates

(1) - X

(2) - Y

(3) - Z

IDID - internal flat for initialization of faces

JDID - special run flag (see input card A)

/INPTT/J, NE, LL, L, NI, NP, NØBJ, IM

J - Number of points $\leq$600

NE - Number of edges $\leq$900

LL - Number of faces $\leq$300

L - Number of points on faces $\leq$1800

NI - Number of intersection points $\leq$1200

NP - Number of intersection points on any one edge $\leq$48

NØBJ - Number of objects (object numbers) $\leq$ 15

IM - Special run flag (see input card F)

/ARCH/PPRP (p, 2), NPP

where p = maximum number of points in perimeter (99)

PPRP - Coordinates of perimeter points

(,1) - X

(,2) - Y

NPP - Number of perimeter points $\leq$99

DIMENSIØN PPRM (2p,2), P (q+2, 2)

Where q = maximum number of intersection points on one line (48)

PPRM - Coordinates of segments on perimeter

(2i-1, 1) - First X of segment i

(2i-1, 2) - First Y of segment i

(2i, 1) - Second X of segment i

(2i, 2) - Second Y of segment i

P - Sorted coordinates of points on a line

(1, 1) - low vertex X

(1, 2) - low vertex Y

(i, 1) - intersection Xi

(i, 2) - intersection Yi

(NP+2, 1) - high vertex X

(NP+2, 2) - high vertex Y

DIMENSIØN XX (s), YY (s), ZZ (s), A (3)

where s = maximum number of points in a solid's base (20)

XX, YY, ZZ } Coordinates of points in a solid's base.

A - Rotation angles of solid (see input card D)

(1) - Azimuth

(2) - Pitch

(3) - Roll

DIMENSIØN IFP (f), Ø(3), SYM (2), AZM (a)

where f = maximum number of points in a face (25)

a = maximum number of input azimuths (card G) (20)

IFP - Point numbers describing a face (card B)

Ø - Offset for input points (card A)

(1) - X

(2) - Y

(3) - Z

SYM - Program Symmetry controls

AZM - Input azimuths (optional, see card G)

/RØTT3/CØSA (3), SINA (3)

CØSA - Cosines of the angles in A

SINA - Sines of the angles in A

/AR/SAREAS (b), NAREAS (b)

where b = maximum number of perimeters in a projection (25)

SAREAS - 2 times area of perimeter

NAREAS - Pointer to first segment in a perimeter

/DEBUGS/IBUG(4)

IBUG - The debug flags (see debug output and cards A and F)

/INSW/ITYPE, ICR, ICL, TYP

(Results from subroutine COVER)

ITYPE - 0 = not covered

1 = on vertex
2 = on edge
3 = covered

(1, 2, 3: assumed covered)

ICR - Number of edge crosses to right of point

ICL - Number of edge crosses to left of point

TYP - Determines if point is hidden (not from cover)

<0, means point is hidden

EXPANSION OF ARRAYS

In order to expand the limits in the program, the dimensions on the arrays given above must be changed. In addition, several program statements must be changed. These statements are listed below.

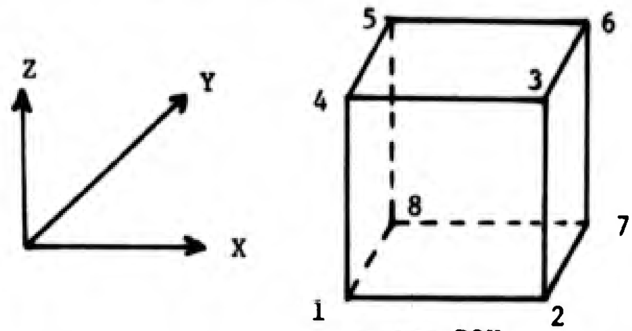
Statement	Sequence
IF(J.GT.600) STØP 6767	270
IF(NØBJ.GE.15) STØP 3333	460
IF(N.GT.25) STØP 2121	560
27 IF(L1.LE.300) GØ TØ 20	630
IF(KR.GT.20) STØP 6736	770
IF(J+KR*K3.GT. 600) STØP 6761	780
CALL COVER (IF, PP, PVRT, 600)	5340
IF (NE.GT.899) STØP 4444	9630

APPENDIX IV

SAMPLE PROBLEMS

(See Following Coding Sheets)

1. This case describes the closed box shown below, using faces only for illustration purposes. The box could have been described by one solid card.



CLOSED BOX

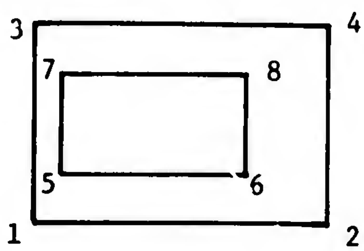
Run F1: Plot at half scale, the perspective and perimeters of the above box rotated and viewed from an elevation of 45 degrees. Calculate areas and converge to .5 square unit. Print intermediate results for each view.

Run F2: Plot at twice scale, the perspectives (only) of the above box rotated and viewed from an elevation of 10 degrees. Calculate areas for two iterations. Print intermediate results for each view.

Run F3: Plot at input scale, three perspectives and perimeters of the above box viewed from an elevation of 30 degrees and at a distance of 250 units (from figure origin). Use azimuth angle of 45, 60, and 330 degrees, respectively, for the three views. Print intermediate results for each view.

In describing the box if the origin was placed so that symmetry was involved, column 41 of card F could be set to X or B. Input is shown in Figure IV-1.

2. This case describes an object with a window, as illustrated below.



FORTRAN CODING FORM

FORM 2265

TITLE

JOB NO. #1- FACES ONLY

Card	Data
CARD A	8 1
B1	0. 0. 0. 5. 0. 0. 5. 0. 4. 0. 0. 4. 0. 3. 4.
B2	5. 3. 4. 6. 3. 0. 0. 3. 0.
C1	4
D1	1 2 3 4
C2	4
D2	3 4 5 6
C3	4
D3	1 2 7 8
C4	4
D4	1 8 5 4
C5	4
D5	2 7 6 3
C6	4
D6	5 6 7 8
E	2
F1	.5 46. .6 1 22
F2	2. 10. -2. 1 12
F3	1. 30. 3 1 250. 22
G	46. 60. 330.
H	
I	
J	

Figure IV-1. Input Data - Faces Only

Card C - Columns 4 - 5 : 10

Card D - Columns 1 - 30: 3 4 2 1 5 6 8 7 5 1

Since the edge that connects points 1 and 5 is given twice, it is considered a pseudo-edge and is not drawn. In effect, it cuts into the hole to connect the hole to the outside.

3. This case demonstrates the use of solids (see plot, Figure IV-2). Input is shown in Figure IV-3.

4. This case demonstrates the use of object types as shown in Figure IV-4. Input is shown in Figure IV-5.

Run F1 defines object 5 (the large solid) to be the target with objects 1 through 4 being shields. Object 6 is a frame.

Run F2 defines objects 2 and 4 as the target with objects 1, 3, and 5 being shields. Object 6 is a frame.

If card E contains a 2 in column 5, the geometry output shown in Figure IV-6 is obtained.

Depending upon the type of input used, the following output is printed.

1 For each face, the following is printed:

Face number (i), Pointer (LFCE), Object number (LFCI).
LFCE (i) points to table LOC (j) (see 4 below).

2 For each point, the following is printed:

Point number (i), Input X, Input Y, Input Z (PVXYZ)

3 For each edge, the following is printed:

Edge number (i), point numbers, face numbers, flag (JEDGE)

4 The following face table is printed:

LOC(j), edge number (JFCE), point number (JFCET).
If FACE i points to j1, and FACE i +1 to j2 (see 1 above), all entries between LOC j1 and LOC j2 - 1 describe FACE i.

5 Minimum and maximum points are printed which describe the extreme corners of a rectangular solid (in input coordinates) which would enclose all the geometry.

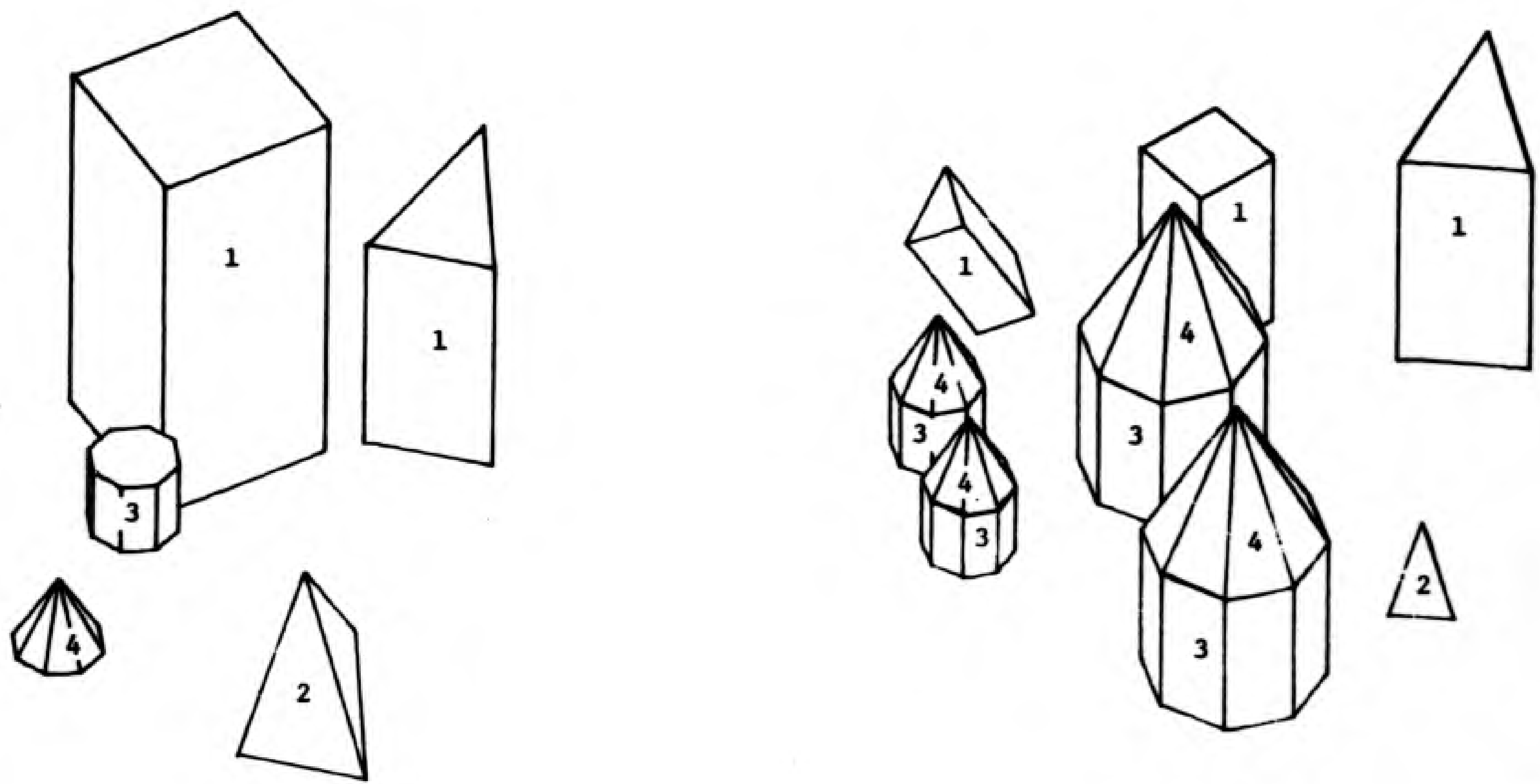


Figure IV-2. Standard Geometrical Shapes

FORM 2265

FORTRAN CODING FORM

TITLE ______ ANALYST ______

JOB NO #3 - SOLIDS ONLY DATE ______ SHEET ______ OF ______

CARD									
A		1	1						
B									
C1	1	3	1						
D1	1.	2.	3.	45.	0.	0.			
D'1	2.	1.	1.	2.	3.				
C2	1	4	2						
D2	4.	1.	1.	45.	0.	0.	.5		
D'2	2.	0.	2.	2.	0.	2.	5.		
C3	2	3	3						
D3	1.	7.	3.	45.	0.	0.	.5		
D'3	2.	1.	1.	2.	1.5	1.5	3.		
C4	3		4						
D4	4.	5.	2.	0.	0.	0.	2.		
D'4	.5	1.							
C5	4								
D5	4.	5.	4.	0.	0.	0.	2.		
D'5	.5	1.							
C6	9	4							
D6	4.	8.	2.	0.	0.	0.			
C7	9	4	5						
D7	7.	5.	2.	0.	0.	0.	.5		
C8	9	1	6						
D8	6.	2.	2.	0.	30.	0.	.5		
E		2							
F	.5			45.		-1.		1	22

Figure IV-3. Input Data - Solids Only

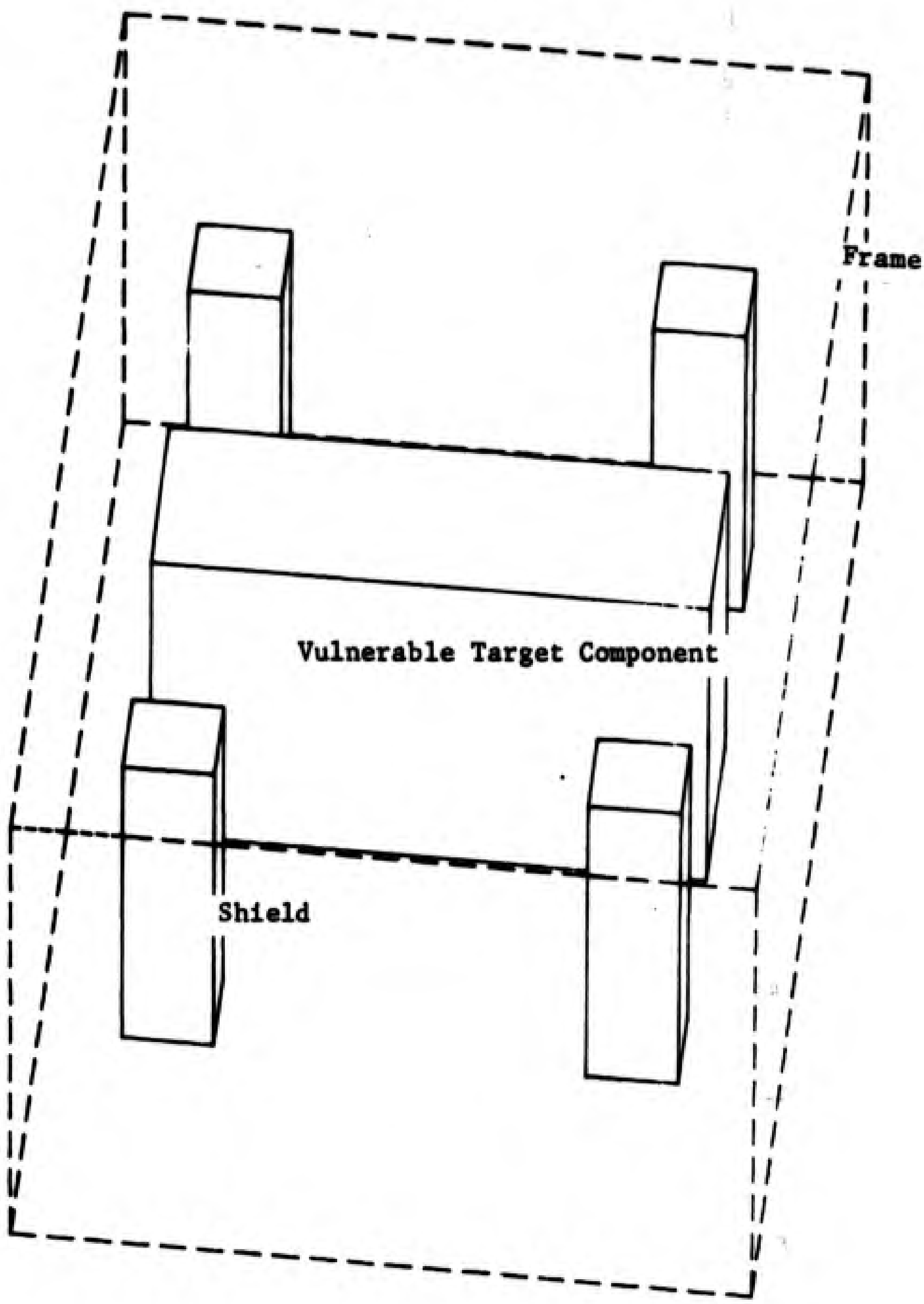


Figure IV-4. Shielding and Identifiable Component Example

FORTRAN CODING FORM

FORM 7265

TITLE ______ ANALYST ______

JOB NO. #4- TEST PROBLEM DATE ______ SHEET ______ OF ______

REMARKS	Col. 1–4	5	6–9	10–14	15–19	20–24	25–29	30–34	40–44	55–61
CARD A		1	1							
B										
C1	1	4		1						
D1	2.		1.	1.						
D'1	1.		0.	1.	1.	0.	1.	4.		
C2	9	1		2						
D2	2.		6.	1.						
C3	9	1		3						
D3	9.		1.	1.						
C4	9	1		4						
D4	9.		6.	1.						
C5	1	4		5						
D5	5.		1.	1.						
D'5	2.		0.	2.	6.	0.	6.	4.		
C6	1	4		6						
D6	0.		0.	0.						
D'6	12.		0.	12.	8.	0.	8.	6.		
E		2								
F1	.5				46.		-1.	1	2	222201
F2	.5				45.		-1.	1	2	202021
H										
I										
J										

Figure IV-5. Input Data - Test Problems

```
 1

1    3    1
1.    2.    3.    45.
2.    1.    1.    2.    3.
2    3    2
3.    4.    3.          45.
2.    1.    .5    2.5   3.
 2
```

FACE	POINTER	OBJECT
1	1	1
2	4	1
3	7	1
4	11	1
5	15	1
6	19	2
7	22	2
8	25	2
9	28	2

POINT	X	Y	Z
1	0.0	0.0	0.0
2	1.00	2.00	3.00
3	1.71	4.12	3.00
4	0.29	4.12	3.00
5	1.00	2.00	6.00
6	1.71	4.12	6.00
7	0.29	4.12	6.00
8	3.00	4.00	3.00
9	4.41	5.00	1.50
10	3.35	6.50	2.65
11	5.12	4.00	0.88

EDGE	PT	PT	FACE	FACE	FLG
1	2	3	1	3	0
2	3	4	1	4	0
3	4	2	1	5	0
4	5	6	2	3	0
5	6	7	2	4	0
6	7	5	2	5	0
7	3	6	3	4	0
8	5	2	3	5	0
9	4	7	4	5	0
10	8	9	6	7	0
11	9	10	6	8	0
12	10	8	6	9	0
13	9	11	7	8	0
14	11	8	7	9	0
15	10	11	8	9	0

Figure IV-6. Geometry Output

LOC	EDGE	POINT
1	1	2
2	2	3
3	3	4
4	4	5
5	5	6
6	6	7
7	1	2
8	7	3
9	4	6
10	8	5
11	2	3
12	9	4
13	5	7
14	7	6
15	3	4
16	8	2
17	6	5
18	9	7
19	10	8
20	11	9
21	12	10
22	10	8
23	13	9
24	14	11
25	11	9
26	15	10
27	13	11
28	12	10
29	14	8
30	15	11

```
********MINIMUM POINT********  ********MAXIMUM POINT********
     0.0         0.0         0.0         5.12       [illegible].50        [illegible].00
```

Figure IV-6. (Concluded)

(The reverse of this page is blank)

APPENDIX V

TARGET DESCRIPTION AND VULNERABILITY PROGRAM

PROGRAM PPLOT CDC 6600 FTN V3.0-P284 OPT=1 04/05/72 12.54.44. PAGE 1

```
             PROGRAM PPLOT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)
       C     PROGRAM PPLOT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,FILMPL)
             COMMON/RAST/N00
             COMMON COEF(300,4)                                                  10100010
            *,LFCE(301),JFCE(1000),PVRT(600,2),PVXYZ(600,3),JEDGE(900,5),        10100020
            *PINT(1200,4),PEYE(3),NMBI(900)                                      10100030
            *,IOBJ(16),INOBJ(15),LFCI(300)                                       10100040
             COMMON /INPTT/J,NE,LL,L,NI,NP,NOBJ,IM                               10100050
             COMMON/JEBUGS/IBUG(4)                                               10100060
             COMMON/ARCH/PPRP(99,2),NPP                                          10100070
             COMMON/AR/SAREAS(25),NAREAS(25)                                     10100080
             COMMON/RNGE/PMX(3),PMN(3),IDIO,JOIO                                 10100090
             COMMON/TEMPY/JFCET(1000),IPNT(100)                                  10100100
             COMMON/ROTT3/COSA(3),SINA(3)                                        10100110
             DIMENSION           IFP(25),O(3),SYM(2),AZM(20)                     10100120
             DIMENSION XX(20),YY(20),ZZ(20),A(3)                                 10100130
             EQUIVALENCE(X,PEYE(1)),(Y,PEYE(2)),(Z,PEYE(3))                      10100140
             DATA SYM/1HO,1HX/                                                   10100150
             DATA PI2/6.283185/                                                  10100160
       C                                                                         10100170
             CALL PLOTCC(0.,0.,-3)                                               10100180
             N00=100                                                             10100190
           5 IDIO=0                                                              10100195
             WRITE(6,1)                                                          10100200
           1 FORMAT(27H1TARGET DESCRIPTION PROGRAM)
       C                                                                         10100210
       C     J=NUMBER OF POINTS                                                  10100220
       C                                                                         10100230
             READ(5,11)J,IBUG,IO,O,FC,JOIO,((PVXYZ(I,M),M=1,3),I=1,J)            10100240
          11 FORMAT(I5,5I1,4F5.0,I5/(15F5.0))                                    10100250
             IF(J.LE.0)GO TO 1000                                                10100260
             IF(J.GT.600)STOP 6767                                               10100270
             IF(FC.LE.0.)FC=1.                                                   10100280
       C     CHECK OFFSET                                                        10100290
             IF(IO.EQ.0)GO TO 15                                                 10100300
             DO 12 I=1,J                                                         10100310
             DO 12 M=1,3                                                         10100320
          12 PVXYZ(I,M)=PVXYZ(I,M)+O(M)                                          10100330
       C                                                                         10100340
       C                                                                         10100350
       C     L1 = FACE NUMBER , NOBJ = OBJECT NUMBER , IOBJ(I) = LOC OF OBJ I    10100360
          15 CONTINUE                                                            10100370
             L1=1                                                                10100380
             NOBJ=0                                                              10100390
       C                                                                         10100400
          20 READ(5,21)K,N,JOBJ                                                  10100410
          21 FORMAT(I1,I4,I5)                                                    10100420
             IOBJ(NOBJ+1)=L1                                                     10100430
             IF(L1.GT.1.AND.JOBJ.EQ.0)GO TO 23                                   10100440
       C     NEW OBJECT                                                          10100450
             IF(NOBJ.GE.15)STOP 3333                                             10100460
             NOBJ=NOBJ+1                                                         10100470
          23 IF(K.EQ.0)GO TO 25                                                  10100480
       C     READ POSITION , ORIENTATION , AND SCALE OF FIGURE                   10100490
             READ(5,36)X1,Y1,Z1,A,FCT                                            10100500
```

```
      IF(FCT.EQ.0.)FCT=1.                                               18100510
      GO TO(30,30,65,65,75,80,30,65,90),K                               18100520
C                                                                       18100530
   25 CONTINUE                                                          18100540
      IF(N.LT.3)GO TO 105                                               18100550
      IF(N.GT.25)STOP 2121                                              18100560
      READ(5,26)(IFP(I),I=1,N)                                          18100570
   26 FORMAT(25I3)                                                      18100580
C                                                                       18100590
      CALL FACES(IFP,N)                                                 18100600
C                                                                       18100610
      L1=L1+1                                                           18100620
   27 IF(L1.LE.300)GO TO 20                                             18100630
C                                                                       18100640
      STOP 5555                                                         18100650
C                                                                       18100660
   30 KR=N+K-1                                                          18100670
C     READ POINTS AND HEIGHT                                            18100680
      READ(5,36)(XX(I),YY(I),I=2,KR),ZP                                 18100690
   36 FORMAT(14F5.0)                                                    18100700
C     SET FIRST POINT                                                   18100710
      XX(1)=X1                                                          18100720
C                                                                       18100730
C     K3=1 -- POINT ON TOP , K3=2 -- RIGHT FIGURE                       18100740
C                                                                       18100750
   38 K3=2+N-KR                                                         18100760
      IF(KR.GT.20)STOP 6736                                             18100770
      IF(J+KR*K3.GT.300)STOP 6761                                       18100780
C     SCALE HEIGHT                                                      18100790
      ZP=ZP*FCT                                                         18100800
C     FIRST POINT                                                       18100810
      YY(1)=Y1                                                          18100820
      ZZ(1)=Z1                                                          18100830
C                                                                       18100840
C     POINT TRANSLATION AND SCALE                                       18100850
      DO 40 I=2,KR                                                      18100860
      YY(I)=FCT*YY(I)+Y1                                                18100870
      XX(I)=FCT*XX(I)+X1                                                18100880
   40 ZZ(I)=Z1                                                          18100890
C     TOP POINT , IF ANY                                                18100900
      IF(K3.EQ.1)ZZ(KR)=Z1+ZP                                           18100910
      JSAVE=J                                                           18100920
      JSV1=J+1                                                          18100930
C     STORE BASE                                                        18100940
C     ROTATE                                                            18100950
      CALL ROTAT3(XX,YY,ZZ,KR,X1,Y1,Z1,A)                               18100960
      DO 43 I=1,KR                                                      18100970
      J=J+1                                                             18100980
      PVXYZ(J,1)=XX(I)                                                  18100990
      PVXYZ(J,2)=YY(I)                                                  18101000
   43 PVXYZ(J,3)=ZZ(I)                                                  18101010
C                                                                       18101020
C     STORE BOTTOM FACE                                                 18101030
      DO 45 I=1,N                                                       18101040
   45 IFP(I)=I+JSAVE                                                    18101050
```

```
      C                                                   10101060
            CALL FACES(IFP,N)                             10101070
      C                                                   10101080
            IF(K3.EQ.1)GO TO 55                           10101090
      C                                                   10101100
      C     CALCULATE AND STORE TOP                       10101110
            ZP1=ZP*SINA(2)                                10101120
            ZP2=ZP*COSA(2)*SINA(3)                        10101130
            ZP3=ZP*COSA(2)*COSA(3)                        10101140
            DO 48 I=1,KR                                  10101150
            J=J+1                                         10101300
            PVXYZ(J,1)=XX(I)+ZP1                          10101310
            PVXYZ(J,2)=YY(I)-ZP2                          10101320
         48 PVXYZ(J,3)=ZZ(I)+ZP3                          10101330
      C                                                   10101340
      C     STORE TOP FACE                                10101350
            DO 50 I=1,N                                   10101360
         50 IFP(I)=IFP(I)+N                               10101370
      C                                                   10101380
            CALL FACES(IFP,N)                             10101390
      C                                                   10101400
      C     STORE SIDE FACES                              10101410
         55 KK=K3+2                                       10101420
            DO 60 M=1,N                                   10101430
            IFP(1)=M+JSAVE                                10101440
            IFP(2)=M+JSV1                                 10101450
            IF(M.EQ.N)IFP(2)=JSV1                         10101460
            IFP(3)=M+JSV1                                 10101470
            IF(K3.EQ.1)GO TO 60                           10101480
            IFP(3)=IFP(2)+N                               10101490
            IFP(4)=IFP(1)+N                               10101500
         60 CALL FACES(IFP,KK)                            10101510
      C                                                   10101520
            L1=L1+N+K3                                    10101530
            GO TO 27                                      10101540
      C                                                   10101550
      C     CIRCULAR BASE(N POINTS)                       10101560
         65 IF(N.EQ.0)N=8                                 10101570
            KR=N+K-3                                      10101580
            PIP=PI2/N                                     10101590
            READ(5,36)RR,ZP                               10101600
            N1=N-1                                        10101610
            DO 68 I=1,N1                                  10101620
            ANG=I*PIP                                     10101630
            XX(I+1)=RR*COS(ANG)                           10101640
         68 YY(I+1)=RR*SIN(ANG)                           10101650
      C     FIRST X VALUE                                 10101660
            XX(1)=X1+RR*FCT                               10101670
            IF(K.EQ.3)GO TO 38                            10101680
      C     SET TOP POINT                                 10101690
            XX(KR)=0.                                     10101700
            YY(KR)=0.                                     10101710
            GO TO 38                                      10101720
      C                                                   10101730
         75 CONTINUE                                      10101740
```

```
   97 CONTINUE
    6 CONTINUE
C
   90 CONTINUE
      IF(N.GT.IOBJ)STOP 2442
C
C     COPY OBJECT N
C
C     JPNT = NO. OF PTS IN THIS FACE (TO DATE)
C     NPNT = NO. OF PTS IN THE OBJECT(TO DATE)
      NPNT=0
      JSV1=J
      IST1=IOBJ(N)
      IST2=IOBJ(N+1)-1
C
C     GET EACH FACE IN OBJECT N
C
      DO 96 I=IST1,IST2
      NSAVE=NPNT
      JPNT=0
      M1=LFCE(I)
      M2=LFCE(I+1)-1
      IF(I.GT.IST1)GO TO 91
C     SAVE FIRST POINT ON FIRST FACE AS REFERENCE POINT
      IFST=JFCET(M1)
C
      XX(1)=X1
      YY(1)=Y1
      ZZ(1)=Z1
C     SET DELTAS (PLUS SCALE)
      DX=X1-PVXYZ(IFST,1)*FCT
      DY=Y1-PVXYZ(IFST,2)*FCT
      DZ=Z1-PVXYZ(IFST,3)*FCT
C
C     GET EACH POINT IN FACE I
   91 CONTINUE
      DO 95 M=M1,M2
      IF(NPNT.EQ.0)GO TO 93
C     SEE IF POINT IS TABLED ALREADY
      DO 92 K3=1,NPNT
      IF(JFCET(M).EQ.IPNT(K3))GO TO 94
   92 CONTINUE
C     NO - STORE POINT
   93 IF(NPNT.GE.100)STOP 1357
      NPNT=NPNT+1
      IPNT(NPNT)=JFCET(M)
      K3=NPNT
   94 JPNT=JPNT+1
   95 IFP(JPNT)=JSV1+K3
C
      M2=NPNT-NSAVE
      IF(M2.EQ.0)GO TO 98
      M3=0
      M1=NSAVE+1
```

```
C
C     TRANSLATE AND SCALE ALL THE NEW POINTS
      DO 96 M=M1,NPNT
      KR=IPNT(M)
      N3=N3+1
      XX(N3)=FCT*PVXYZ(KR,1)+DX
      YY(N3)=FCT*PVXYZ(KR,2)+DY
   96 ZZ(N3)=FCT*PVXYZ(KR,3)+DZ
C
      CALL ROTAT3(XX,YY,ZZ,N3,X1,Y1,Z1,A)
C
      DO 97 M=1,N3
      J=J+1
      PVXYZ(J,1)=XX(M)
      PVXYZ(J,2)=YY(M)
   97 PVXYZ(J,3)=ZZ(M)
C
C     STORE FACE
   98 CALL FACES(IFP,JPNT)
C
C
      L1=L1+IST2-IST1+1
      GO TO 27
C
C
C
C     A R C H I T E C T U R A L   P R I N T   O U T
C
C
  105 CONTINUE
      LFCE(LL+1)=L+1
      IF(JDID.NE.0)GO TO 109
C     FIND RANGE OF X,Y,Z
      DO 108 M=1,3
      PMX(M)=PVXYZ(1,M)
      PMN(M)=PVXYZ(1,M)
      DO 108 I=2,J
      IF(PMX(M).LT.PVXYZ(I,M))PMX(M)=PVXYZ(I,M)
      IF(PMN(M).GT.PVXYZ(I,M))PMN(M)=PVXYZ(I,M)
  108 CONTINUE
  109 CONTINUE
      IF(N.LT.2)GO TO 150
      WRITE(6,110)(I,LFCE(I),LFCI(I),I=1,LL)
  110 FORMAT(//7X25HFACE     POINTER    OBJECT/(3I10))
      WRITE(6,120)(I,(PVXYZ(I,M),M=1,3),I=1,J)
  120 FORMAT(//6H POINT,6X1HX,9X1HY,9X1HZ/(I5,3F10.2))
      WRITE(6,125)(I,(JEDGE(I,M),M=1,5),I=1,NE)
  125 FORMAT(//2X29HEDGE  PT   PT  FACE FACE  FLG/(6I5))
      WRITE(6,130)(I,JFCE(I),JFCET(I),I=1,L)
  130 FORMAT(//3X14HLOC EDGE POINT/(3I5))
      WRITE(6,135)PMN,PMX
  135 FORMAT(60H0********MINIMUM POINT******** ********MAXIMUM POINT****
     1****/6F10.2)
C
  150 CONTINUE
```

```
      READ(5,134)P1,P2,PHI,CONV,JR,IM,IREA,R,SY,IPLOT,IBUG,P3,INOBJ      18102850
      WRITE(6,134)P1,P2,PHI,CONV,JR,IM,IREA,R,SY,IPLOT,IBUG,P3,INOBJ     18102860
  134 FORMAT(2F5.2,2F10.3,I2,2I1,F6.2,A1,5I1,1X,F6.2,2X,15I1)            18102870
      IF(P1.EQ.0.)GO TO 5                                                18102880
      IF(P3.NE.0.)CALL PLOTCC(P3,0.,-3)                                  18102890
      IF(IM.NE.0)GO TO 205                                               18102900
C     CHECK SYMMETRY                                                     18102910
      DO 160 ISY=1,2                                                     18102920
      IF(SY.EQ.SYM(ISY))GO TO 170                                        18102930
  160 CONTINUE                                                           18102940
      ISY=4                                                              18102950
  170 CONTINUE                                                           18102960
      I6=6                                                               18102970
      IF(CONV.GE.0.)GO TO 182                                            18102980
C     ITERATION OPTION                                                   18102990
      I6=.5-CONV                                                         18103000
      CONV=.5                                                            18103010
  182 CONV=P1*P1*CONV                                                    18103020
      IF(R.EQ.0.)R=1000.                                                 18103030
      PHY=PHI/57.2958                                                    18103040
      Z=R*SIN(PHY)                                                       18103050
      Q=R*COS(PHY)                                                       18103060
      IF(IBUG(1).GT.1)WRITE(6,186)R,Z,CONV                               18103070
  186 FORMAT(9H0RANGE =,F9.2,4H Z =,F9.2,7H CONV =,F8.4)                 18103080
      AREAS=0.                                                           18103090
      NK=1                                                               18103100
      ARQ=1.                                                             18103110
      N=ISY                                                              18103120
      NN=N                                                               18103130
      IF(ISY.LT.4)NN=N+1                                                 18103140
      IF(ISY. EQ.1)ARQ=2.                                                18103150
      TOT=ISY*1.570796                                                   18103160
      IF(JR.LE.0)GO TO 200                                               18103170
C     AZMUTH INPUT                                                       18103180
      READ(5,196)(AZM(I),I=1,JR)                                         18101160
      WRITE(6,196)(AZM(I),I=1,JR)                                        18101170
  196 FORMAT(8F10.4)                                                     18101180
      I6=1                                                               18101190
      NN=JR                                                              18101200
C                                                                        18101210
C                                                                        18101220
  200 NU=NN                                                              18101230
  205 DO 350 I=1,I6                                                      18101240
      II=I
      DTH=TOT/N                                                          18101250
C                                                                        18101260
      DO 300 M=NK,NN,NK                                                  18101270
      IF(IM.NE.0)GO TO 260                                               18101280
C     AREA FACTOR                                                        18101290
      ARF=ARQ                                                            18103190
      IF(ISY.EQ.4)GO TO 230                                              18103200
      IF(NK.EQ.2)GO TO 225                                               18103210
      IF(ISY.EQ.1)GO TO 227                                              18103220
      IF(M.NE.2)GO TO 230                                                18103230
  225 ARF=2.*ARQ                                                         18103240
```

```
      227 NO=NG+ARF-1                                                  10103250
      230 NTH=1                                                        10103260
          TH=M*DTM-DTM+.1                                              10103270
          IF(JR.GT.0)TH=AZM(M)/57.2958                                 10103280
      250 X=Q*COS(TH)                                                  10103290
          Y=Q*SIN(TH)                                                  10103300
          CALL INIT33(-P1,0,X,Y,Z,PMN(1),PMX(1),PMN(2),PMX(2),PMN(3),PMX(3))10103310
      260 CALL ARCH3(AREA,IPLOT,IREA)                                  10103320
          IF(IREA.EQ.0)GO TO 300                                       10103330
          IF(AREA.GE.0.)GO TO 280                                      10103340
          IF(IBUG(1).GT.0)WRITE(6,271)TH,AREA,NTH                      10103350
      271 FORMAT(19H0TROUBLE IN AREA AT,2F10.4,I5)                     10103360
          IF(JR.GT.0)GO TO 270                                         10103370
          TH=TH+.1*DTM                                                 10103380
          NTH=NTH+1                                                    10103390
          IF(IBUG(1).GT.1 .AND. IPLOT.NE.0)CALL PLOTCC(P2,0.,-3)       10103400
          IF(NTH.LE.3)GO TO 250                                        10103410
      270 NO=NO-1                                                      10103420
          GO TO 300                                                    10103430
      280 AREAS=AREAS+AREA*ARF                                         10103440
          IF(IBUG(1).GT.1)WRITE(6,285)TH,X,Y,AREA,AREAS,M              10103450
      285 FORMAT(11H0AZIMUTH = ,F8.4,4H X =,F9.2,4H Y =F9.2,7H AREA =,F9.4, 10103460
         * 6H SUM =,F9.4,11H VIEW NO. =,I3)                            10103470
          IF(IPLOT.LE.1)GO TO 300                                      10103480
C         PLOT PERIMETER                                               10103490
          CALL PLOTCC(P2,0.,-3)                                        10103500
          NIA=1                                                        10103510
          IP=3                                                         10103520
          DO 295 M1=1,NPP
          IF(IP.EQ.3)GO TO 291
          IX1=N00*PPRP(M1-1,1)
          IY1=N00*PPRP(M1-1,2)
          IX2=N00*PPRP(M1,1)
          IY2=N00*PPRP(M1,2)
          CALL LINEV(IX1,IY1,IX2,IY2)
      291 CONTINUE
          IP=2                                                         10103550
          IF(M1.LT.NAREAS(NIA))GO TO 295                               10103560
          IP=3                                                         10103570
          NIA=NIA+1                                                    10103580
      295 CONTINUE                                                     10103590
C                                                                      10103600
      300 IF(IPLOT.NE.0)CALL PLOTCC (P2,0.,-3)                         10103610
          IF(AREAS.EQ.0.)GO TO 150                                     10103620
C                                                                      10103630
          AREAA=AREAS/NO                                               10103640
          IF(IBUG(1).GT.0)WRITE(6,356)I,AREAA                          10103650
          IF(I6.EQ.1)GO TO 355                                         10103660
          NO=NO+N                                                      10103670
          N=2*N                                                        10103680
          NK=2                                                         10103690
          IPLOT=0                                                      10103700
          NN=N                                                         10103710
          IF(I.EQ.1)GO TO 350                                          10103720
          IF(ABS(AREAA-AREA0).LE.CONV)GO TO 355                        10103730
```

```
  350 AREAO=AREAA                                              10103740
C                                                              10103750
  355 AREAA=FC*FC*AREAA/P1/P1                                  10103760
      WRITE(6,356)II,AREAA                                     10103770
  356 FORMAT(5H0AFTER,I4,23H ITERATIONS, THE AREA =,F8.3//)    10103780
      GO TO 150                                                10103790
C                                                              10103800
 1000 CONTINUE                                                 10103810
      STOP 77                                                  10103830
      END                                                      10103840
```

```
      SUBROUTINE ARCH3(AREA,IPLOT,IREA)
C     THREE D ARCHITCTURAL PLOTTING ROUTINE                             10103850
C       (ALSO FINDS PERIMETER AND AREA SEEN)                            10103860
C                                                                       10103870
C     TABLES ARE GENERATED IN MAIN AND PASSED THRU COMMON               10103880
C                                                                       10103890
C     IF IREA.NE.0, AREA WILL BE CALCULATED.                            10103900
C     IF IPLOT.NE.0 PLOT WILL BE MADE                                   10103910
      COMMON COEF(300,4)                                                10103920
     *,LFCE(301),JFCE(1800),PVRT(600,2),PVXYZ(600,3),JEDGE(900,5),
     *PINT(1200,4),PEYE(3),NMBI(900)
     *,IOBJ(16),INOBJ(15),LFCI(300)
      COMMON /INPTT/J,NE,LL,L,NI,NP,NOBJ,IM
      COMMON/DEBUGS/IBUG(4)                                             10103990
      COMMON /ANSW/ITYPE,ICR,ICL,TYP                                    10104000
      COMMON/ARCH/PPRP(99,2),NPP                                        10104010
      COMMON/AR/SAREAS(25),NAREAS(25)                                   10104020
      DIMENSION PC(3),AL(3),BL(3),PP(2),PPRM(199,2),P(50,2)             10104030
      NSEG=0                                                            10104040
      IF(IM.NE.0)GO TO 150                                              10104050
C                                                                       10104060
C     TRANSFORM POINTS                                                  10104070
      DO 136 M=1,J                                                      10104080
      CALL XFORM(PVXYZ(M,1),PVXYZ(M,2),PVXYZ(M,3),PVRT(M,1),PVRT(M,2))  10104090
  136 CONTINUE                                                          10104100
      IF(IBUG(2).GT.0)WRITE(6,141)(M,(PVRT(M,N) ,N=1,2),M=1,J)          10104110
  141 FORMAT(6H0POINT,7X,1HX,9X,1HY/(I5,2F10.4))                        10104120
C                                                                       10104130
C     GET ALL INTERSECTIONS                                             10104140
      CALL INTER                                                        10104150
C                                                                       10104160
C                                                                       10104170
C     DRAW EACH EDGE                                                    10104180
C                                                                       10104190
  150 CONTINUE                                                          10104200
      DO 300 M=1,NE                                                     10104210
      ICK=0                                                             10104220
C     CHECK PROPERTY OF OBJECT ON THIS EDGE.                            10104230
C     INOBJ(I) = 0 - OBJECT I IS OPAQUE AND TARGET                      10104240
C              = 1 - OBJECT I IS TRANSPARENT FRAME                      10104250
C              = 2 - OBJECT I IS OPAQUE AND SHIELD                      10104260
C              = 3 - OBJECT I IS IGNORED                                10104270
      DO 200 I0=3,4                                                     10104280
C     FACE ON EDGE                                                      10104290
      I01=JEDGE(M,I0)                                                   10104300
      IF(I01.EQ.0)GO TO 200                                             10104310
C     OBJECT ON FACE                                                    10104320
      I02=LFCI(I01)                                                     10104330
C     SEE IF TO BE USED                                                 10104340
      IF(INOBJ(I02)-2)210,190,200                                       10104350
C     SHIELD. IF EDGE ALSO USED ON TARGET OR FRAME USE AS SUCH.         10104360
  190 ICK=I02                                                           10104370
  200 CONTINUE                                                          10104380
C     IGNORE IF NOT SHIELD                                              10104390
      IF(ICK.EQ.0)GO TO 300                                             10104400
                                                                        10104410
```

```
      IB2=ICK                                                           18104420
C     SKIP CUTS (PSEUDO EDGES)                                          18104430
  210 CONTINUE                                                          18104440
      IP2=2                                                             18104450
      IF(JEDGE(M,5).GT.0)IP2=3                                          18104460
      K1=JEDGE(M,1)                                                     18104470
      K2=JEDGE(M,2)                                                     18104480
      K3=K1                                                             18104490
      K4=K2                                                             18104500
      DX=PVRT(K1,1)-PVRT(K2,1)                                          18104510
      IF(ABS(DX).LE..0001)GO TO 215                                     18104520
      IF(DX)220,215,217                                                 18104530
  215 IF(PVRT (K1,2) -  PVRT (K2,2)  )220,300,217                       18104540
  217 K4=K1                                                             18104550
      K3=K2                                                             18104560
C                                                                       18104570
C     LENGTH OF LINE                                                    18104580
C                                                                       18104590
  220 DO 222 N1=1,2                                                     18104600
      BL(N1)=PVRT(K4,N1)-PVRT(K3,N1)                                    18104610
      AL(N1)=PVXYZ(K4,N1)-PVXYZ(K3,N1)                                  18104620
  222 CONTINUE                                                          18104630
      AL(3)=PVXYZ(K4,3)-PVXYZ(K3,3)                                     18104640
C                                                                       18104650
C                                                                       18104660
C                                                                       18104670
      NP=0                                                              18104680
C     ANY INTERSECTIONS                                                 18104690
      IF(NI.LE.0)GO TO 250                                              18104700
      IF(NMBI(M).EQ.0)GO TO 250                                         18104710
C     YES                                                               18104720
      N1=1                                                              18104730
      DO 240 N=1,NI                                                     18104740
  230 I1=PINT(N,N1)+.5                                                  18104750
      IF(M-I1)232,235,240                                               18104760
  232 GO TO(233,240),N1                                                 18104770
  233 N1=2                                                              18104780
      GO TO 230                                                         18104790
C                                                                       18104800
C     YES - ADD POINT TO SORTED LIST                                    18104810
C                                                                       18104820
  235 CALL SORTH(PINT(N,3),PINT(N,4),P(2,1),P(2,2),NP,48)               18104830
      IF(NP.EQ.NMBI(M))IF(NP-48)250,241,241                             18104840
  240 CONTINUE                                                          18104850
  241 STOP 7654                                                         18104860
C                                                                       18104870
C     PUT VERTICES IN LIST                                              18104880
  250 NP=NP+2                                                           18104890
      DO 252 N1=1,2                                                     18104900
      P(1,N1)=PVRT(K3,N1)                                               18104910
  252 P(NP,N1)=PVRT(K4,N1)                                              18104920
      IF(IBUG(2).GT.2)WRITE(6,253)NP,M,((P(N,N1),N1=1,2),N=1,NP)        18104930
  253 FORMAT(I5,15H POINTS ON EDGE,I4/(6(2F8.4,4X)))                    18104340
C                                                                       18104950
C                                                                       18104960
```

```
      C     DRAW SEGMENT BY SEGMENT - STARTING FROM LOW VERTEX
      C
            N1S=0
            DO 280 N1=1,NP
            IPER=0
      C     REVERSE PERIMETER IF TYPE 2
            IF(INOBJ(IO2).EQ.2)IPER=1
            DIS0=993.
            DIS2=993.
            IP=3
            IF(N1.EQ.1)IF(IPLOT)275,280,275
      C
      C     SEE IF SAME POINT
            DO 255  N2=1,2
            IF(P(N1,N2).NE.P(N1-1,N2))GO TO 257
        255 CONTINUE
            GO TO 280
      C
      C
      C     CENTER OF SEGMENT
        257 DO 258 N2=1,2
        258 PP(N2)=(P(N1,N2)+P(N1-1,N2))/2.
      C
            DO 260 N2=1,3
            N3=1
            IF(BL(1).LE..01)N3=2
        260 PC(N2)=PVXYZ(K3,N2)+AL(N2)*(PP(N3)-P(1,N3))/BL(N3)
      C
      C     IS IT HIDDEN BY A FACE
      C
            DO 265 IF=1,LL
      C     CHECK FACE FOR TRANSPARENCY
            I03=LFCI(IF)
            IF(INOBJ(I03).EQ.1.OR.INOBJ(I03).EQ.3)GO TO 265
      C     OPAQUE (WILL HIDE WHAT IS BEHIND IT)
      C     SKIP FACES ON EDGE
            IF(JEDGE(M,3).EQ.IF .OR.JEDGE(M,4).EQ.IF)GO TO 265
            CALL COVER(IF,PP,PVRT,600)
            IF(IBUG(2).GT.6)WRITE(6,262)PP,ITYPE,ICR,ICL,IF,I02,I03
            IF(ITYPE.EQ.0)GO TO 265
      C
      C     COVERED
      C
      C     INVERT PERIMETER, IF TYPE 2
            IF(INOBJ(I02).EQ.2)GO TO 259
            IF(IPLOT.EQ.0)GO TO 280
      C     TYPE 0 COVERED BY TYPE 2 MAY STILL BE ON PERIMETER
            IF(INOBJ(I02).EQ.0.AND.INOBJ(I03).EQ.2)GO TO 259
            IPER=1
        259 CONTINUE
            T1=COEF(IF,4)
            T2=T1
            DO 263 N2=1,3
            T1=T1+COEF(IF,N2)*PC(N2)
        263 T2=T2+COEF(IF,N2)*PEYE(N2)
```

```
      IF(IBUG(2).GT.5)WRITE(6,262)PP,ITYPE,ICR,ICL,IF,M,N1,PC,T1,T2
  262 FORMAT(2F10.4,6I5,3F10.4,2E15.4)
      IF( ABS(T2).LT..005)GO TO 375
      TYP=T1/T2
      IF(TYP.LT.0.)GO TO 270
C
C     NOT HIDDEN BY THIS CHECK FACE
      IF(INOBJ(IB2).NE.2)GO TO 265
C     TYPE 2 EDGE
      IF(INOBJ(IB3).EQ.2)GO TO 266
C     COVERED BY TYPE 0 FACE (SEGMENT MAY BE ON PERIMETER)
      IPER=0
C     DISTANCE FROM THIS TYPE 2 SEGMENT TO NEAREST TYPE 0 FACE THAT
C             COVERS IT.
      DIS0=AMIN1(DIS0,TYP)
      GO TO 265
C     DISTANCE FROM THIS TYPE 2 SEGMENT TO NEAREST TYPE 2 FACE THAT
C             COVERS IT
  266 DIS2=AMIN1(DIS2,TYP)
C
  265 CONTINUE
C
C     NO NOT HIDDEN
C
      IF(IREA.EQ.0 )GO TO 269
      IF(IPER.NE.0)GO TO 269
C     NO PERIMETER FOR TYPE 1
      IF(INOBJ(IB2).EQ.1)GO TO 269
C     NO PERIMETER FOR TYPE 2 UNLESS COVERING A TYPE 0 FACE WITH NO
C             TYPE 2 FACE BETWEEN
      IF(INOBJ(IB2).EQ.2)IF(DIS2-DIS0)269,269,268
C     NOT COVERED. SAVE SEGMENT FOR PERIMETER
  268 IF(NSEG.GE.199)STOP 7766
      N4=1
      IF(N1.GT.N1S)GO TO 267
      N4=2
      NSEG=NSEG-2
C
  267 DO 261 N3=N4,2
      DO 261 N2=1,2
  261 PPRM(NSEG+N3,N2)=P(N1+N3-2,N2)
      NSEG=NSEG+2
      N1S=N1+1
C
C
  269 IP=IP2
  270 IF(IPLOT.EQ.0)GO TO 280
      IF(INOBJ(IB2).NE.1)GO TO 275
C     FRAME
      IF(IP.EQ.3)GO TO 275
      CALL DASHLN(P(N1-1,1),P(N1-1,2),0.,0.,2.)
      IP=3
  275 CALL PLOTOD(P(N1,1),P(N1,2),IP)
C
  280 CONTINUE
```

```
      C
        300 CONTINUE
      C
      C
            IF(IREA.EQ.0)RETURN
            IF(IBUG(2).GT.1)WRITE(6,361)NSEG,((PPRM(N,M),M=1,2),N=1,NSEG)
      C
      C     REMOVE ANY DUPLICATE SEGMENTS
            DO 304 N4=1,NSEG,2
            N3=N4+2
        295 CONTINUE
            IF(PPRM(N3,1).LT.-900)GO TO 302
            DO 301 N2=1,2
            IF(PPRM(N3,N2).NE.PPRM(N4,N2))GO TO 303
        301 CONTINUE
            IV=N3+1
            IF(2*(N3/2).EQ.N3)IV=N3-1
            DO 299 N2=1,2
            IF(PPRM(IV,N2).NE.PPRM(N4+1,N2))GO TO 303
        299 CONTINUE
      C     FOUND
            PPRM(IV,1)=-999
            PPRM(N3,1)=-999
            N3=MIN0(IV,N3)
        302 N3=N3+1
        303 N3=N3+1
            IF(N3.LE.NSEG)GO TO 295
        304 CONTINUE
      C
      C     INITIALIZE PERIMETER SEARCH
      C
            NPP=0
            SUM=0.
            IA=0
            M1=1
      C     GENERATE PERIMETER POINTS STARTING FROM POINT M1
      C
        305 IV=M1+1
            DO 310 N=1,2
            NPP=NPP+1
            DO 310 M=1,2
        310 PPRP(NPP,M)=PPRM(M1+N-1,M)
            AREA=(PPRM( M1,1)-PPRM(IV,1))*(PPRM( M1,2)+PPRM(IV,2))
      C
            N=NPP+1
      C     SEARCH FOR POINT .EQ. IV
        320 DO 330 N1=3,NSEG
            IF(N1.EQ.IV)GO TO 330
            DO 325 N2=1,2
            IF(PPRM(IV,N2) .NE. PPRM(N1,N2))GO TO 330
        325 CONTINUE
      C     FOUND
            PPRM(IV,1)=-999.
            PPRM(N1,1)=-999.
            IV=N1+1
```

```
      IF(2*(N1/2).EQ.N1)IV=N1-1
      JDONE=1
C     PUT POINT IN PERIMETER AND SEE IF DONE
      DO 328 N2=1,2
      IF(PPRM(IV,N2) .NE. PPRM(M1,N2))GO TO 328
      JDONE=JDONE+1
  328 PPRP(N,N2)=PPRM(IV,N2)
      AREA=(PPRP(N-1,1)-PPRP(N,1))*(PPRP(N-1,2)+PPRP(N,2))+AREA
      GO TO(350,350,355),JDONE
  330 CONTINUE
      GO TO 365
  350 IF(IBUG(2).GT.4)WRITE(6,351)M1,N,N1,IV,AREA,SUM
  351 FORMAT(4I5,2F10.4)
      N=N+1
      GO TO 320
C
C     SAVE AREA AND SEGMENT POINTER
C
  355 CONTINUE
      NPP=N
      IA=IA+1
      IF(IA.GT.25)GO TO 365
      SAREAS(IA)=ABS(AREA)
      NAREAS(IA)=NPP
      PPRM(M1,1)=-999.
      PPRM(IV,1)=-999.
      IF(IBUG(2).LT.4)GO TO 358
      WRITE(6,361)NSEG,((PPRM(N,M),M=1,2),N=1,NSEG)
      WRITE(6,361)NPP ,((PPRP(N,M),M=1,2),N=1,NPP)
C     FIND NEXT AREA (IF, ANY)
C
  358 DO 359 M1=3,NSEG,2
      IF(PPRM(M1,1).GT.-900.)GO TO 305
  359 CONTINUE
C
      IF(IBUG(2).GT.1)WRITE(6,357)(M,SAREAS(M),NAREAS(M),M=1,IA)
  357 FORMAT(/5X3HNO.,5X4HAREA,5X7HPOINTER/(I7,F12.4,I8))
C
C     SUM AREA
C
      DO 364 N=1,IA
      AREA=SAREAS(N)
      IF(IA.EQ.1)GO TO 364
      M1=NAREAS(N)
      DO 360 M=1,2
  360 PP(M)=(PPRP(M1,M)+PPRP(M1-1,M))/2.
  361 FORMAT(/6HOPERIM,I5/(6(2F8.4,4X)))
      DO 362 M=1,IA
      IF(M.EQ.N)GO TO 362
      CALL COVER(-M,PP,PPRP,99)
      IF(ITYPE.NE.0)GO TO 363
  362 CONTINUE
      GO TO 364
C     COVERED
  363 AREA=-AREA
```

```
      364 SUM=SUM+AREA
C                                                                       10107170
C                                                                       10107180
C         AREA                                                          10107190
          AREA=.5*SUM                                                   10107200
C                                                                       10107210
          IF(IBUG(2).GT.0)WRITE(6,361) NPP,((PPRP(N,M),M=1,2),N=1,NPP)  10107220
          RETURN                                                        10107230
      365 AREA=-1.                                                      10107240
          IF(IBUG(2).LE.1)RETURN                                        10107250
          WRITE(6,361)N,((PPRP(N2,M),M=1,2),N2=1,N)                     10107260
          WRITE(6,366)M1,N1,IV,NPP,IA,(PPRM(IV,N2),PPRM(M1,N2),N2=1,2)  10107270
      366 FORMAT(5I10,4F15.8)                                           10107280
          RETURN                                                        10107290
      375 AREA=- ABS(T2)-2.                                             10107300
          RETURN                                                        10107310
          END                                                           10107320
                                                                        10107340
```

```
      SUBROUTINE INTER                                              18107350
      COMMON COEF(300,4)
     *,LFCE(301),JFCE(1800),PVRT(600,2),PVXYZ(600,3),JEDGE(900,5),
     *PINT(1200,4),PEYE(3),NMBI(900)
     *,IOBJ(16),INOBJ(15),LFCI(300)
      COMMON /INPTT/J,NE,LL,L,NI,NP,NOBJ,IM                         18107410
      COMMON/DEBUGS/IBUG(4)                                         18107420
      COMMON /ANSW/ITYPE,ICR,ICL,TYP                                18107430
C                                                                   18107440
C     J - NUMBER OF POINTS                                          18107450
C     NE- NUMBER OF EDGES                                           18107460
C     LL- NUMBER OF FACES                                           18107470
C     L - LENGTH OF JFCE TABLE                                      18107480
C     NI- NUMBER OF INTERSECTIONS                                   18107490
C                                                                   18107500
C     PINT(NI,1) = EDGE NUMBER I                                    18107510
C     PINT(NI,2) = EDGE NUMBER K                                    18107520
C     PINT(NI,3) = X INTERSECT                                      18107530
C     PINT(NI,4) = Y INTERSECT                                      18107540
C                                                                   18107550
C     NMBI(I) = NUMBER OF INTERSECTIONS ON EDGE I                   18107560
C                                                                   18107570
      IF(IBUG(4).GT.0)WRITE(6,20)                                   18107580
   20 FORMAT(//5X1HX7X1HY6X3HXI15X3HYI15X3HXI25X3HYI25X3HSLI5X3HXK15X 18107590
     * 3HYK15X3HXK25X3HYK25X3HSLK4X3HIGG3X1HI4X1HK4X1HN)            18107600
      NI=0                                                          18107610
      DO 30 I=1,NE                                                  18107620
   30 NMBI(I)=0                                                     18107630
C                                                                   18107640
      DO  90 I=2,NE                                                 18107650
      LI1=JEDGE(I,1)                                                18107660
      LI2=JEDGE(I,2)                                                18107670
      XI1=PVRT(LI1,1)                                               18107680
      XI2=PVRT(LI2,1)                                               18107690
      YI1=PVRT(LI1,2)                                               18107700
      YI2=PVRT(LI2,2)                                               18107710
      DXI=XI2-XI1                                                   18107720
      IF(DXI.NE.0.)SLI=(YI2-YI1)/DXI                                18107730
      XMINI=AMIN1(XI1,XI2)                                          18107740
      XMAXI=AMAX1(XI1,XI2)                                          18107750
      YMINI=AMIN1(YI1,YI2)                                          18107760
      YMAXI=AMAX1(YI1,YI2)                                          18107770
C                                                                   18107780
      I1=I-1                                                        18107790
C                                                                   18107800
C     FIND INTERSECTIONS WITH EDGE I                                18107810
      DO 80  K=1,I1                                                 18107820
C                                                                   18107830
C     SEE IF EDGE I IS ON SAME FACE AS EDGE K                       18107840
C     SEE IF EDGE I IS ON SAME POINT AS EDGE K                      18107850
      DO 50 M=3,4                                                   18107860
      IF(JEDGE(I,M).EQ.0)GO TO 51                                   18107870
      DO 50 N=3,4                                                   18107880
      IF(JEDGE(I,M).EQ.JEDGE(K,N))GO TO 80                          18107890
      IF(JEDGE(I,M-2).EQ.JEDGE(K,N-2))GO TO 80                      18107900
```

```
         50 CONTINUE
C           NO
C           GET VERTICES
         51 LK1=JEDGE(K,1)
            LK2=JEDGE(K,2)
C
            XK1=PVRT(LK1,1)
            XK2=PVRT(LK2,1)
C           CHECK RANGE OF X
            XMK=AMAX1(XK1,XK2)
            IF(XMINI.GT.XMK)GO TO 80
            XNK=AMIN1(XK1,XK2)
            IF(XMAXI.LT.XNK)GO TO 80
C
            YK1=PVRT(LK1,2)
            YK2=PVRT(LK2,2)
C           CHECK RANGE OF Y
            YMK=AMAX1(YK1,YK2)
            IF(YMINI.GT.YMK)GO TO 80
            YNK=AMIN1(YK1,YK2)
            IF(YMAXI.LT.YNK)GO TO 80
C
C           CHECK FOR COMMON END POINT
C
            ADX=ABS(XMK-XMAXI)-.00001
            ADY=ABS(YMK-YMAXI)-.00001
            IF(ADX.LE.0..AND.ADY.LE.0.)GO TO 80
            ADX=ABS(XNK-XMINI)-.00001
            ADY=ABS(YNK-YMINI)-.00001
            IF(ADX.LE.0..AND.ADY.LE.0.)GO TO 80
C
C           FIND INTERSECTION
C
            DXK=XK2-XK1
            ADXI=ABS(DXI)-.0001
            ADXK=ABS(DXK)-.0001
            IF(ADXI.LE.0.)IF(ADXK)80,80,65
            IF(ADXK.LE.0.)GO TO 70
C
            SLK=(YK2-YK1)/DXK
            IF(SLI.EQ.SLK)GO TO 80
C           IGG=1 - NEITHER VERTICAL
C               2 - I LINE IS VERTICAL
C               3 - K LINE IS VERTICAL
            IGG=1
C           X
            X=(YI1-YK1-SLI*XI1+SLK*XK1)/(SLK-SLI)
C           IS X ON BOTH LINES
         56 IF((X-XI1)*(X-XI2).GT.0.)GO TO 80
            Y=SLI*(X-XI1)+YI1
            IF(IGG.EQ.3)IF((Y-YK1)*(Y-YK2))58,80,80
C
         57 IF((X-XK1)*(X-XK2).GT.0.)GO TO 80
C           YES  -  SAVE POINT
         58 NI=NI+1
```

```
      PINT(NI,1)=I                                                      18100460
      PINT(NI,2)=K                                                      18100470
      PINT(NI,3)=X                                                      18100480
      PINT(NI,4)=Y                                                      18100490
      NMBI(I)=NMBI(I)+1                                                 18100500
      NMBI(K)=NMBI(K)+1                                                 18100510
      GO TO 79                                                          18100520
C                                                                       18100530
   65 X=XI1                                                             18100540
      IGG=2                                                             18100550
      SLK=(YK2-YK1)/DXK                                                 18100560
      Y=SLK*(X-XK1)+YK1                                                 18100570
      IF((Y-YI1)*(Y-YI2))57,57,60                                       18100580
C                                                                       18100590
   70 X=XK1                                                             18100600
      IGG=3                                                             18100610
      GO TO 55                                                          18100620
C                                                                       18100630
   79 CONTINUE                                                          18100640
      IF(IBUG(N).GT.0)                                                  18100650
     *WRITE(  ,81)X,PINT(NI,4),XI1,YI1,XI2,YI2,SLI,XK1,YK1,XK2,YK2,SLK, 18100660
     *  IGG,I,K,NI                                                      18100670
   81 FORMAT(12F8.2,4I5)                                                18100680
C                                                                       18100690
   80 CONTINUE                                                          18100700
   90 CONTINUE                                                          18100710
      RETURN                                                            18100720
      END                                                               18100730
```

```
SUBROUTINE  ROTAT3                                      CDC 6600 FTN V3.0-P284 OPT=1  04/05/72  12.54.44.       PAGE    1

              SUBROUTINE ROTAT3(X,Y,Z,N,XZ,YZ,ZZ,A)                                10100740
              DIMENSION X(1),Y(1),Z(1),A(3)                                        10100750
              COMMON/ROTT3/COSA(3),SINA(3)                                         10100760
              DO 10 I=1,3                                                          10100770
              B=A(I)/57.29578                                                      10100780
              COSA(I)=COS(B)                                                       10100790
           10 SINA(I)=SIN(B)                                                       10100800
              T11=COSA(1)*COSA(2)                                                  10100810
              T21=SINA(1)*COSA(3)+COSA(1)*SINA(2)*SINA(3)                          10100820
              T31=SINA(1)*SINA(3)-COSA(1)*SINA(2)*COSA(3)                          10100830
              T12=-SINA(1)*COSA(2)                                                 10100840
              T22=COSA(1)*COSA(3)-SINA(1)*SINA(2)*SINA(3)                          10100850
              T32=COSA(1)*SINA(3)+SINA(1)*SINA(2)*COSA(3)                          10100860
              T13=SINA(2)                                                          10100870
              T23=-COSA(2)*SINA(3)                                                 10100880
              T33=COSA(2)*COSA(3)                                                  10100890
              DO 20 I=1,N                                                          10100900
              X1=X(I)-XZ                                                           10100910
              Y1=Y(I)-YZ                                                           10100920
              Z1=Z(I)-ZZ                                                           10100930
              X(I)=X1*T11+Y1*T12+Z1*T13+XZ                                         10100940
              Y(I)=X1*T21+Y1*T22+Z1*T23+YZ                                         10100950
              Z(I)=X1*T31+Y1*T32+Z1*T33+ZZ                                         10100960
           20 CONTINUE                                                             10100970
              RETURN                                                               10100980
              END                                                                  10100990
```

```
      SUBROUTINE FACES(IFP,N)                                           18109000
C     DEVELOPE TABLES FOR FACE L CONSISTING OF THE N POINTS INDICATED BY18109010
C        IFP                                                            18109020
      COMMON COEF(300,4)
     *,LFCE(301),JFCE(1800),PVRT(600,2),PVXYZ(600,3),JEDGE(900,5),
     *PINT(1200,4),PEYE(3),NMBI(900)
     *,IOBJ(16),INOBJ(15),LFCI(300)
      COMMON /INPTT/J,NE,LL,L,NI,NP,NOBJ,IM                             18109090
      COMMON/DEBUGS/IBUG(4)                                             18109100
      COMMON/RNGE/PMX(3),PMN(3),IJID,JDID                               18109110
      COMMON/TEMPY/JFCET(1800),IPNT(100)
      DIMENSION IC(3),CF(2,3),IFP(1)                                    18109130
      IF(IDID.EQ.1234)GO TO 10                                          18109140
C     INITIALIZE                                                        18109150
      IDID=1234                                                         18109160
C     K=FACE POINTER                                                    18109170
      K=1                                                               18109180
C     NE=NUMBER OF EDGES                                                18109190
      NE=0                                                              18109200
C     LL=FACE NUMBER                                                    18109210
      LL=0                                                              18109220
C                                                                       18109230
      IF(IBUG(3).GT.0)WRITE(6,136)                                      18109240
  136 FORMAT(130H *********    C O E F F I C I E N T   M A T R I X   ****18109250
     1*********    F A C E   E Q U A T I O N S   ****)                  18109260
C                                                                       18109270
   10 LL=LL+1                                                           18109280
      LFCE(LL)=K                                                        18109290
C     NOBJ=OBJECT NUMBER                                                18109300
      LFCI(LL)=NOBJ                                                     18109310
      DO 15 I=1,N                                                       18109320
      L=K+I-1                                                           18109330
      JFCET(L)=IFP(I)                                                   18109340
   15 JFCE(L)=IFP(I)                                                    18109350
C                                                                       18109360
C     JEDGE(I,J)-EDGE I , J=1,5                                         18109370
C              1 - POINT ON EDGE                                        18109380
C              2 - POINT ON EDGE                                        18109390
C              3 - FACE ON EDGE                                         18109400
C              4 - FACE ON EDGE                                         18109410
C              5 - PSEUDO EDGE FLAG                                     18109420
C                                                                       18109430
      JFCE(L+1)=JFCE(K)                                                 18109440
      DO 40 M=K,L                                                       18109450
C     POINT NUMBERS OF EDGE                                             18109460
      IF(JFCE(M).GT.J)STOP 1234                                         18109470
      IF(NE.EQ.0)GO TO 35                                               18109480
C                                                                       18109490
C     HAS EDGE BEEN COUNTED                                             18109500
      DO 33 I1=1,NE                                                     18109510
      DO 32 I2=1,2                                                      18109520
      IJ=JFCE(M+I2-1)                                                   18109530
      DO 31 I3=1,2                                                      18109540
      IF(IJ.EQ.JEDGE(I1,I3))GO TO 32                                    18109550
   31 CONTINUE                                                          18109560
```

```
            GO TO 33
         32 CONTINUE
            GO TO 36
         33 CONTINUE
      C
      C     NEW EDGE TO ADD TO LIST
            IF(NE.GT.899)STOP 4444
         35 NE=NE+1
            JEDGE(NE,1)=JFCE(M)
            JEDGE(NE,2)=JFCE(M+1)
            JEDGE(NE,3)=LL
            JEDGE(NE,4)=0
            JEDGE(NE,5)=0
            JFCE(M)=NE
            GO TO 40
      C
      C
         36 CONTINUE
            IF(LL.EQ.JEDGE(I1,3))JEDGE(I1,5)=1
            JEDGE(I1,4)=LL
            JFCE(M)=I1
         40 CONTINUE
      C
      C     FIND COEFFICIENTS OF FACE
      C
            IC3=IFP(3)
            DO 60 I=1,2
            IC12=IFP(I)
            DO 60 M=1,3
         60 CF(I,M)=PVXYZ(IC3,M)-PVXYZ(IC12,M)
      C
            COEF(LL,1)=CF(1,2)*CF(2,3)-CF(1,3)*CF(2,2)
            COEF(LL,2)=CF(1,3)*CF(2,1)-CF(1,1)*CF(2,3)
            COEF(LL,3)=CF(1,1)*CF(2,2)-CF(1,2)*CF(2,1)
            COEF(LL,4)=0.
            DO 70 I=1,3
         70 COEF(LL,4)=COEF(LL,4)-COEF(LL,I)*PVXYZ(IC3,I)
            IF(IBUG(3).GT.0)WRITE(6,71)CF,(COEF(LL,I),I=1,4),LL
         71 FORMAT(10F10.2,I5)
            IF(N.EQ.3)GO TO 90
      C
      C     CHECK EACH POINT ON FACE FOR COPLANARITY
            DO 80 KL=4,N
            IP4=IFP(KL)
            T1=COEF(LL,4)
            DO 75 M=1,3
         75 T1=T1+COEF(LL,M)*PVXYZ(IP4,M)
            IF( ABS(T1).LT..005)GO TO 80
            WRITE(6,77)LL,IP4,T1
         77 FORMAT(15H ERROR IN PLANE,I3,7H, POINT,I4,F10.4)
         80 CONTINUE
      C
         90 K=L+1
            LFCE(LL+1)=K
            RETURN
            END
```

```
      SUBROUTINE COVER(IF,PC,PVRT,JJ)                                   18110130
      DIMENSION PC(1),PVRT(JJ,2)                                        18110150
      COMMON COEF(300,4)
     *,LFCE(301),JFCE(1800),DUMY(600,2),PVXYZ(600,3),JEDGE(900,5),
     *PINT(1200,4),PEYE(3),NMBI(900)
     *,IOBJ(16),INOBJ(15),LFCI(300)
      COMMON/JEBUGS/IBUG(4)                                             18110200
      COMMON /ANSW/ITYPE,ICR,ICL,TYP                                    18110210
      COMMON/AR/SAREAS(25),NAREAS(25)                                   18110220
C                                                                       18110230
C     CHECK TO SEE IF FACE (IF) COVERS POINT (PC)                       18110240
C     IF.LT.0 INDICATES PROJECTION -IF RATHER THAN FACE                 18110250
C     ITYPE = 0 - NOT COVERED                                           18110260
C           = 1 - ON VERTEX                                             18110270
C           = 2 - ON EDGE                                               18110280
C           = 3 - COVERED                                               18110290
C                                                                       18110300
      IF(IF.GT.0)GO TO 10                                               18110310
      LF=1                                                              18110320
      IF(IF.LT.-1)LF=NAREAS(-IF-1)+1                                    18110330
      LSTV=NAREAS(-IF)-1                                                18110340
      GO TO 15                                                          18110350
C                                                                       18110360
   10 LF=LFCE(IF)                                                       18110370
      LSTV=LFCE(IF+1)-1                                                 18110380
C                                                                       18110390
C     LF = POINTER TO FACE                                              18110400
C                                                                       18110410
   15 PX=PC(1)                                                          18110420
      PY=PC(2)                                                          18110430
C                                                                       18110440
C     PX,PY = POINT COORDINATES                                         18110450
C                                                                       18110460
C     ICR , ICL -- NUMBER OF LINES TO RIGHT , LEFT OF POINT             18110470
      ICR=0                                                             18110480
      ICL=0                                                             18110490
C                                                                       18110500
C                                                                       18110510
      DO 100 I=LF,LSTV                                                  18110520
      IVFE=JFCE(I)                                                      18110530
      IF(I.EQ.LF)GO TO 25                                               18110540
C     NEW VALUES                                                        18110550
      IVF1=IVF2                                                         18110560
      DX=DXN                                                            18110570
      DY=DYN                                                            18110580
      IVF2=IVF1+1                                                       18110590
      GO TO 40                                                          18110600
C                                                                       18110610
   25 IF(IF.GT.0)GO TO 30                                               18110620
      IVF1=LF                                                           18110630
      IVF2=LF+1                                                         18110640
      GO TO 35                                                          18110650
C                                                                       18110660
   30 IVF1=JEDGE(IVFE,1)                                                18110670
      IVFN=JFCE(LF+1)                                                   18110680
```

```
      IF(IVF1.EQ.JEDGE(IVFN,1).OR.IVF1.EQ.JEDGE(IVFN,2))IVF1=JEDGE(IVFE,
     *2)
   35 DX=PX-PVRT(IVF1,1)
      DY=PY-PVRT(IVF1,2)
C     SEE IF ON VERTEX
      IF(DX.EQ.0.)IF(DY)40,200,40
C     NO
C
   40 IF(IF.LT.0)GO TO 42
      IVF2=JEDGE(IVFE,2)
      IF(IVF2.EQ.IVF1)IVF2=JEDGE(IVFE,1)
   42 DXN=PX-PVRT(IVF2,1)
      DYN=PY-PVRT(IVF2,2)
C     SEE IF ON VERTEX
      IF(DXN.EQ.0.)IF(DYN)45,200,45
C
C     SEE IF HORIZONTAL EDGE
   45 IF(PVRT(IVF1,2).EQ.PVRT(IVF2,2))IF(DY)100,140,100
C     NO
C     SEE IF HORIZONTAL LINE THRU P INTERSECTS EDGE
      IF(DYN*DY)50,48,100
C     INTERSECTS VERTEX - COUNT OR NOT
   48 IF(DYN.GT.0. .OR.DY.GT.0.)GO TO 100
C     YES
C     SLOPE OF EDGE (Y AXIS)
   50 SL= PVRT(IVF2,1)-PVRT(IVF1,1)
      SL=SL/(PVRT(IVF2,2)-PVRT(IVF1,2))
C     DELTA X TO EDGE
      DXL=SL*DY-DX
C     SEE IF ON EDGE
C           OR IF CROSSES LEFT OR RIGHT OF POINT
C
      IF(DXL)97,300,99
   97 ICL=ICL+1
      GO TO 100
   99 ICR=ICR+1
  100 CONTINUE
C
C
C     EVEN MEANS NOT COVERED
      IF(2*(ICR/2).EQ.ICR)GO TO 150
      ITYPE=3
      RETURN
C
C     SEE IF ON OR OFF HORIZONTAL EDGE
  140 IF(DX*DXN)300,200,100
C
  150 ITYPE=0
      RETURN
C
  200 ITYPE=1
      RETURN
C
  300 ITYPE=2
      RETURN
      END
```

```
      SUBROUTINE SORTH(X1,X2,TAB1   ,TAB2   ,I,IMAX)              18111250
      DIMENSION TAB1(1),TAB2(1)                                   18111260
      IF(I.EQ.0) GO TO 33                                         18111270
      IX=MIN0(IMAX,I)                                             18111280
      DO 30 J1=1,IX                                               18111290
      J=IX+1-J1                                                   18111300
      IF(X1-TAB1(J))25,20,34                                      18111310
   20 IF(X2-TAB2(J))25,34,34                                      18111320
   25 TAB1(J+1)=TAB1(J)                                           18111330
      TAB2(J+1)=TAB2(J)                                           18111340
   30 CONTINUE                                                    18111350
   33 J=0                                                         18111360
   34 TAB1(J+1)=X1                                                18111370
      TAB2(J+1)=X2                                                18111380
      I=I+1                                                       18111390
      RETURN                                                      18111400
      END                                                         18111410
```

```
      SUBROUTINE PLOTDD(X,Y,I)
      COMMON/RAST/N00                                        10111420
      DATA XP,YP,IP/0.,0.,0/
      IF(I.NE.2)GO TO 10                                     10111430
      IF(X.EQ.XP.AND.Y.EQ.YP)RETURN                          10111440
      IX1=N00*XP+.5                                          10111450
      IY1=N00*YP+.5                                          10111460
      IX2=N00*X+.5
      IY2=N00*Y+.5
      CALL LINEV(IX1,IY1,IX2,IY2)
      IP=0                                                   10111470
      GO TO 15                                               10111480
   10 IP=1                                                   10111490
   15 XP=X                                                   10111500
      YP=Y                                                   10111510
      RETURN                                                 10111520
      END                                                    10111530
                                                             10111540
```

```
      SUBROUTINE PLOTCC(X,Y,IP)
      IF(IP.LT.0)GO TO 10
      RETURN
   10 CALL FRAMEV(0)
      RETURN
      END
```

```
      SUBROUTINE DASHLN(XI,YI,XF,YF,DI)
      REAL LENGTH
      DIMENSION XI(1),YI(1)
C  THIS DEFINES NORMAL INTERVAL
      DATA APROX1,XMULT1/.0625,15./
C  INITIALIZE FOR PEN 1
      IPENS=2
      XMULT=XMULT1
      APROX=APROX1
      XORG=0.0
C  MOVE TO THE INITIAL POINT WITH PEN UP
   69 CALL PLOTOO(XI(1)+XORG,YI(1),3)
C  INITIALIZE FINAL POINT OF FIRST LINE
      XBAR=XF
      YBAR=YF
C  TEST FOR TYPE OF LINE
      IF(DI)1,2,3
C SOLID LINES PROCESSOR
 1    XBAR=XI(2)
      YBAR=YI(2)
C  SOLID LINE PROCESSOR
 2    LINES = ABS(DI+1.)
      DO 200 L=1,LINES
      CALL PLOTOO(XBAR,YBAR,2)
      XBAR=XI(L+2)
 200  YBAR=YI(L+2)
      RETURN
C DASH LINE PROCESSOR
 3    M=3
C TEST FOR NUMBER AND TYPE OF DASHED LINES
      IF(DI-1.) 4,5,6
C DASHED LINE, 1 TO 1 RATIO
  4   M=4
C  DASHED LINE 2 TO 1 RATIO
 5    LINES=1
      GO TO 50
C  DASHED LINE 2 TO 1 RATIO
 6    XBAR=XI(2)
      YBAR=YI(2)
      LINES = DI-1.
C  BEGIN PLOTTING LINES
 50   DO 7 L=1,LINES
C  COMPUTE LENGTH OF LINE TO BE PLOTTED
      DELTAX = XBAR-XI(L)
      DELTAY = YBAR-YI(L)
       LENGTH = SQRT(DELTAX ** 2 + DELTAY ** 2 )
C TEST FOR MINIMUM PASSIBLE DASHED LINE
 1000 IF(LENGTH.GE.FLOAT(M+2)*APROX) GO TO 13
      APROX = APROX/2.
      XMULT=((XMULT+1.)*2.) +1.
      GO TO 1000
C  SET UP TO DRAW SOLID LINE
 9999 CONTINUE
      X=XBAR
      Y=YBAR
```

```
      CALL PLOTOD(X,Y,2)
      GO TO 14
C  COMPUTE APPROXIMATE NUMBER OF EQUAL INTERVALS
 13   NUM = LENGTH * XMULT
C  TEST FOR START AND STOP WITH DASH
 8    IF(MOD(NUM-2,M).EQ.0) GO TO 9
      NUM=NUM+1
      GO TO 8
C  REFINE BASIC INTERVAL LENGTH
 9    XNUM=NUM
      DX=DELTAX/XNUM
      DY=DELTAY/XNUM
C INITIALIZE FIRST END POINT
      X=XI(L)
      Y=YI(L)
C  INITIALIZE RATIO COUNTER
      DO 12 I=1,NUM
C PLOT LOOP, COMPUTE TERMINAL POINT
      X=X+DX
      Y=Y+DY
C  LOWER PEN
      IPEN=IPENS
C  TEST TO RAISE PEN
      J=I
      IF(M.EQ.3) GO TO 11
C EQUAL RATIO
      IF(MOD(J,M).NE.0) J=J+1
 11   IF(MOD(J,M).EQ.0) IPEN=3
      XORG=0.
C  PLOT THE LINE
      CALL PLOTOD(X+XORG,Y,IPEN)
C  INCREMENT RATIO COUNTER
 12   CONTINUE
C  TEST FOR END OF LINE
 14   XBAR= XI(L+2)
 7    YBAR=YI(L+2)
      RETURN
      END
```

```
      SUBROUTINE INIT3D(XPAG,YPAG,XO,YO,ZO,XN,XMAX,YN,YMAX,ZMIN,ZMAX)
      COMMON/PARAM/XL,YL,XMIN,YMIN,TDIS,DIS,QX,QY,QZ,COSA,COSB,COSC,
     1   OX,OY,OZ,SINC,PX,PY,IQUAD,DUM,XLEN,YLEN
      DIMENSION XX(2),YY(2),ZZ(2)
      ISW = 0
      OX=XO
      OY=YO
      OZ=ZO
      XMIN=XN
      YMIN=YN
      XX(1) = XMIN
      XX(2) = XMAX
      YY(1) = YMIN
      YY(2) = YMAX
      ZZ(1) = ZMIN
      ZZ(2) = ZMAX
      PX = XMIN+.5*(XMAX-XMIN)
      PY = YMIN+.5*(YMAX-YMIN)
      PZ = ZMIN+.5*(ZMAX-ZMIN)
      DIS = SQRT((PX-OX)**2+(PY-OY)**2+(PZ-OZ)**2)
      COSA = (PX-OX)/DIS
      COSB = (PY-OY)/DIS
      COSC = (PZ-OZ)/DIS
      SINC = SQRT(1.-COSC**2)
      TDIS = DIS/2.
      IF(XPAG.GT.0.)GO TO 23
      TDIS=-XPAG*DIS
      ISW=1
   23 XL = 10000.
      YL = 10000.
      XH = -10000.
      YH = -10000.
      QX = OX + TDIS*COSA
      QY = OY + TDIS*COSB
      QZ = OZ + TDIS*COSC
      DO 1 I = 1,2
      X = XX(I)
      DO 1 J = 1,2
      Y = YY(J)
      DO 1 K = 1,2
      Z = ZZ(K)
      FK = TDIS/((X-OX)*COSA + (Y-OY)*COSB + (Z-OZ)*COSC)
      FL = OX + FK*(X-OX)
      FM = OY + FK*(Y-OY)
      FN = OZ + FK*(Z-OZ)
      XP    = ((FL-QX)*COSB - (FM-QY)*COSA)/SINC
      YP    = (FN-QZ)/SINC
      IF (XP.LT.XL) XL=XP
      IF (XP.GT.XH) XH=XP
      IF (YP.LT.YL) YL=YP
      IF (YP.GT.YH) YH=YP
    1 CONTINUE
      IF (ISW.EQ.1) GO TO 24
C
C     ADJUST POIS SO THAT PLOT FILLS DESIRED AREA
```

```
      C
            XLL = XH - XL
            YLL = YH - YL
            RATIO = XLL/XPAG
            TYLL = YLL/RATIO
            IF (TYLL.GT.YPAG) RATIO=YLL/YPAG
            TDIS = TDIS/RATIO
            ISW = 1
            GO TO 23
      C
      C
         24 XL=-XL
            YL=-YL
            XLEN = XMAX-XMIN
            YLEN = YMAX-YMIN
            IQUAD=0
            IF(OX.GE.XMAX)IQUAD=IQUAD+2
            IF(OY.GE.YMAX)IQUAD=IQUAD+1
            RETURN
            END
```

```
CDC   SUBROUTINE PLOTXYZ(X,Y,Z,IPEN)
      SUBROUTINE PLTXYZ(X,Y,Z,IPEN)
      CALL XFORM(X,Y,Z,PX,PY)
      CALL PLOTCC(PX,PY,IPEN)
      RETURN
      END
```

```
      SUBROUTINE XFORM(X,Y,Z,OX,OY)
      COMMON/PARAM/XL,YL,XMIN,YMIN,TDIS,DIS,QX,QY,QZ,COSA,COSB,COSC,
     1   OX,OY,OZ,SINC,PX,PY,IQUAD,DUM,XLEN,YLEN
      FK = TDIS/((X-OX)*COSA + (Y-OY)*COSB + (Z - OZ)*COSC)
      FL = OX + FK*(X-OX)
      FM = OY + FK*(Y-OY)
      FN = OZ + FK*(Z   -OZ)
      OX   = ((FL-QX)*COSB - (FM-QY)*COSA)/SINC + XL
      OY   = (FN-QZ)/SINC + YL
      RETURN
      END
```

REFERENCES

1. Air Force Armament Laboratory, R&D Statement of Work DLRV 72-22 dated 30 July 1971.

2. Lerman, H. N., "The Three Dimensions of Architectural Design Automation," Proceedings of the SHARE, ACM, IEEE Design Automation Workshop, June 28 - 30, 1971.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Martin Marietta Corporation Orlando, Florida	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

TARGET DESCRIPTION AND VULNERABILITY PROGRAM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Report - 17 January 1972 to 19 May 1972

5. AUTHOR(S) (First name, middle initial, last name)

George W. Brooks
Harvey N. Lerman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1972	196	2

8a. CONTRACT OR GRANT NO F08635-72-C-0103 b. PROJECT NO 2549 c. Task No. 01 d. Work Unit No. 006	9a. ORIGINATOR'S REPORT NUMBER(S) OR 11,972 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFATL-TR-72-129

10. DISTRIBUTION STATEMENT

Distribution limited to U. S. Government agencies only; this report documents test and evaluation; distribution limitation applied June 1972. Other requests for this document must be referred to the Air Force Armament Laboratory (DLRV), Eglin Air Force Base, Florida 32542.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
Available in DDC	Air Force Armament Laboratory Air Force Systems Command Eglin Air Force Base, Florida 32542

13. ABSTRACT

This program provided the Air Force an existing target description computer program which was demonstrated on the Eglin Air Force Base CDC 6600 computer by completing two check runs. The subsequent formulation, coding, and incorporation of two improvements in the program which provided a shielding capability and the input of standard geometrical shapes resulted in a modified program being adopted for use with the Eglin Air Force Base computer. The revised program was demonstrated by computing the presented area of the fuel tanks of a generic fighter aircraft, and making perspective plots of the target with hidden lines removed, using the SC4020 plotter. This target description and vulnerability program also provides the Air Force with a unique tool for making rapid descriptive and vulnerability assessments of targets through the use of high speed computers and automatic plotting equipment.

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Target Description Computer Program						
Target Descriptive and Vulnerability Assessment						