NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Contract NGR 14-008-002

STRUCTURAL DESIGN CONCEPTS FOR FUTURE SPACE MISSIONS

Progress Report for period
May 1, 1968 to October 31, 1968

THE SCHOOL OF TECHNOLOGY
SOUTHERN ILLINOIS UNIVERSITY

STRUCTURAL DESIGN
CONCEPTS FOR FUTURE
SPACE MISSIONS

National Aeronautics
and Space Administration
November 1, 1968

Progress Report
NASA Contract
NGR 14-008-002

Julian H. Lauchner
R. Buckminster Fuller
Joseph D. Clinton
Mark B. Mabee
Richard M. Moeller
Richard Flood

NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

FOREWARD

This report was prepared in the School of Technology at Southern Illinois University under National Aeronautics and Space Administration Contract No. NGR 14-008-002.

This report covers the progress of work from May 1, 1968 to October 31, 1968.

Personnel participating in the research include Julian H. Lauchner, principal investigator, R. Buckminster Fuller, Joseph D. Clinton, Mark B. Mabee, Richard M. Moeller, and Richard Flood.

ABSTRACT

This report explains one method of subdividing a polyhedron into triangular facets and "exploding" it into the surface of a sphere.

A mathematical model is included which explains the geometry used in subdividing and transforming the icosahedron into the structural sphere. Also included are a computer program and a plot routine used in the computations.

TABLE OF CONTENTS

INTRODUCTION

This report explains one method of subdividing a polyhedron into triangular facets and "exploding" it onto the surface of a sphere. A structure is thereby given which may be used in spherical form.

The tetrahedron, octahedron or icosahedron are the fundamental geometrical configurations of the structure. A further subdivision of the configuration chosen is accomplished by subdividing each principle side of each principle polyhedral triangle into any number of segments The order of subdivision is determined by subdividing the triangle and the origin (or center) of the polyhedron into equal angle segments, using the origin as the vertice for subdivision. The points of intersection of the equal angle segments with the principle side determines the subdivision along the principle side of the principle polyhedral triangle.

The points of subdivision on each side of the Principle polyhedral triangle are connected with line segments which are parallel to the two remaining sides of the principle polyhedral triangle under consideration. They intersect at a number of points which define a triangular grid of subdivision. Due to the method of subdivision, small triangular "windows" occur in the grid. The centers of these windows are found by one of two methods (discussed later in this report) and are used as the vertices of a triangular grid of subdivision of the principle polyhedral face and are then transformed to the surface of the sphere which circumscribes the polyhedron. The cords that connect these transformed vertices thus define the structural grid that comprise the structural configuration desired.

A mathematical model has been determined which explains the geometry used in subdividing and transforming the polyhedron into the structural configuration desired. From this model a computer program has been written which gives the necessary information needed for construction and analysis of the structure. As a further aid in investigation of the various forms, a plot routine was developed to give a graphical output of each of the structural forms.

BASIC GEOMETRY

DEFINITIONS:

DIHEDRAL ANGLE (β) an angle formed by two planes meeting in a common line. The planes themselves are the faces of the dihedral angle, and the common line is the element. To measure the dihedral angle measure the angle whose vertex is on the element of the dihedral angle and whose sides are perpendicular to the element and lie one in each face of the dihedral angle.

FACE ANGLE (α) an angle formed by two elements meeting in a common point and lying in a plane that is one of the faces of the polyhedron.

AXIAL ANGLE (Ω) an angle formed by an element and a radius from the center of the polyhedron meeting in a common point and the vertex of the axial angle sharing a vertex of the polyhedron.

PRINCIPLE ICOSAHEDRAL TRIANGLE (PIT) any one of the 20 equal equilaterial triangles which forms the face of the regular icosahedron.

PRINCIPLE SIDE any one of the three sides of the principle icosahedral triangle.

FREQUENCY the number of equal parts into which a principle side is subdivided.

FACES the triangles making up the "exploded" structural form.

GEOMETRY

This program works with a tetrahedron, octahedron, or icosahedron circumscribed by a unit sphere. The icosahedron was chosen as an example to illustrate the geometry of the program. The icosahedron is oriented in a three dimensional rectangular coordinate system so that the vertices of one principle icosahedral triangle are:

$$(x, y, z) = \left(0, \frac{\sqrt{\tau}}{\sqrt[4]{5}}, \frac{1}{\sqrt[4]{5}\sqrt{\tau}}\right)$$

$$\simeq (0, .850651, .525731)$$

$$(x, y, z) = \left(\frac{1}{\sqrt[4]{5}\sqrt{\tau}}, 0, \frac{\sqrt{\tau}}{\sqrt[4]{5}}\right)$$

$$\simeq (.525731. 0. .850651)$$

$$(x, y, z) = \left(\frac{\sqrt{\tau}}{\sqrt[4]{5}}, \frac{1}{\sqrt[4]{5}\sqrt{\tau}}, 0\right)$$

$$\simeq (850651, .525731, 0)$$

$$\text{where } \tau = \frac{1 + \sqrt{5}}{2}$$

with the intersections of the axis X, Y, Z, located at the origin (0,0,0) of the icosahedron, Figure 1.

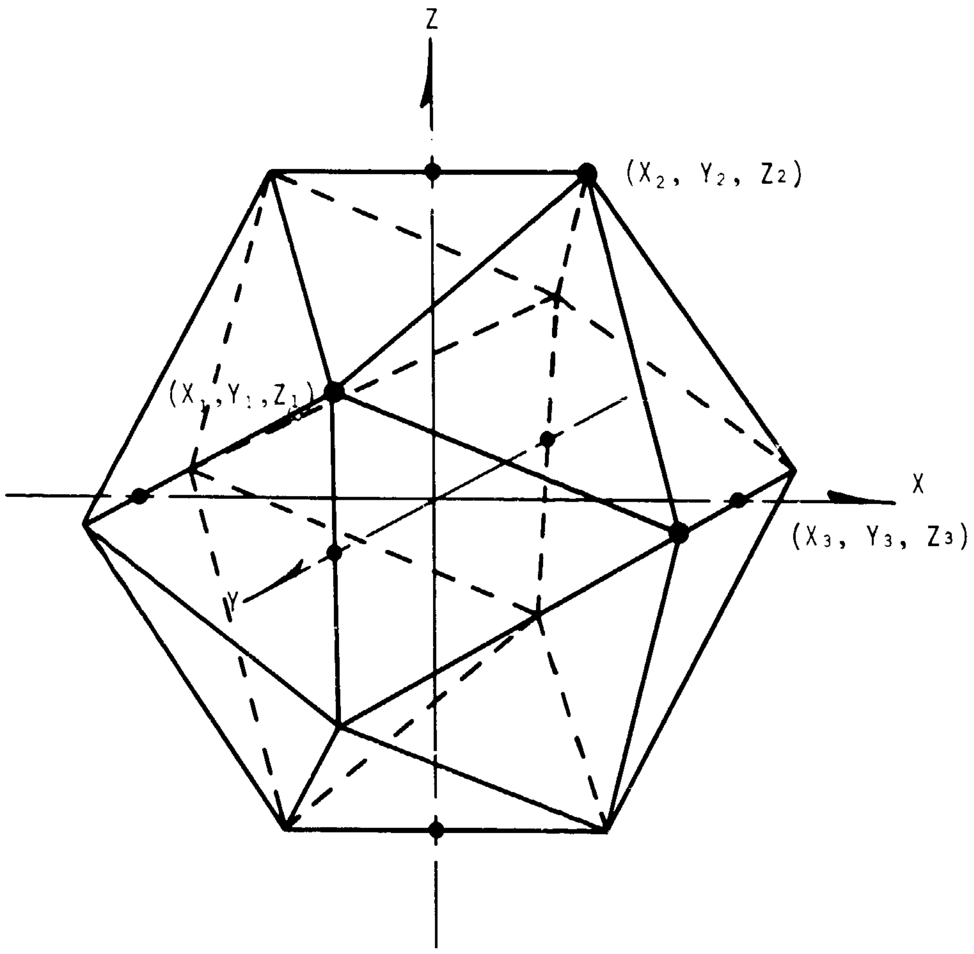


Figure 1

This principle icosahedral triangle is divided into smaller triangular units which are "exploded" onto the surface of a sphere constituting the desirable space form.

Using the following formula the planes consisting of the edges of the PIT and the origin (X_1, Y_1, Z_1) (X_2, Y_2, Z_2) (X_3, Y_3, Z_3) are rotated from 3-space into 2-space, Figure 2.

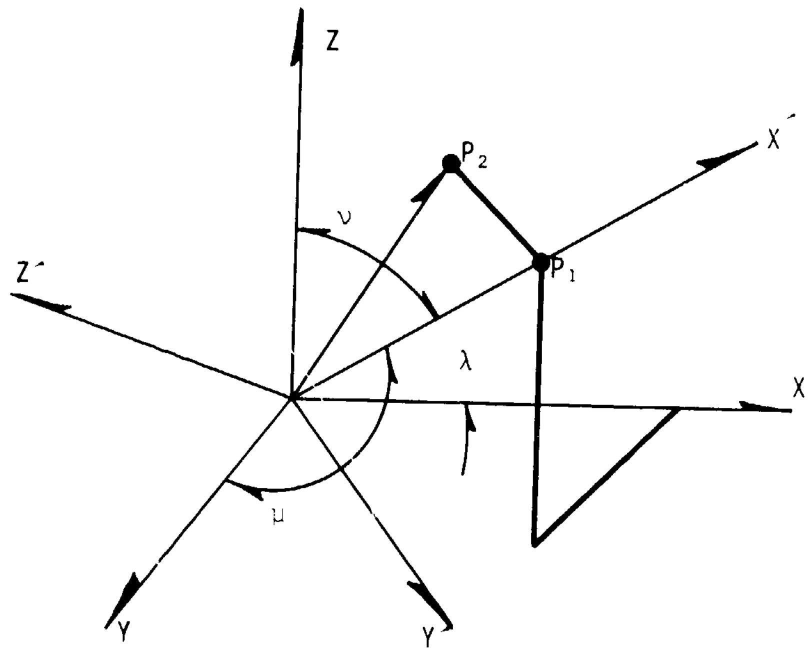


Figure 2

$$x' = \lambda_1 x + \mu_1 y + \nu_1 z$$
$$y' = \lambda_2 x + \mu_2 y + \nu_2 z \qquad [1]$$
$$z' = \lambda_3 x + \mu_3 y + \nu_3 z$$

Where λ, μ, ν are direction cosines of the X´-axis, Y´-axis, and Z´-axis respectively with respect to the old axis and are found by:

$$\lambda_1 = x_1/\sqrt{x_1^2 + y_1^2 + z_1^2}$$
$$\mu_1 = y_1/\sqrt{x_1^2 + y_1^2 + z_1^2}$$
$$\nu_1 = z_1/\sqrt{x_1^2 + y_1^2 + z_1^2}$$

λ_2, λ_3: μ_2, μ_3; and ν_2, ν_3 are found similarly.

The edge of the PIT is subdivided into units by the following method, Figures 3 and 4.

FIND: the angle ϕ contained within the rotated triangle consisting of $\overline{P_1P_2}$, and the origin with the vertex located at the origin.

$$\phi = \text{Arctan}\left(\frac{P_{y2}}{P_{x2}}\right) \cdot r \qquad [2]$$

where r = 1 and is considered constant

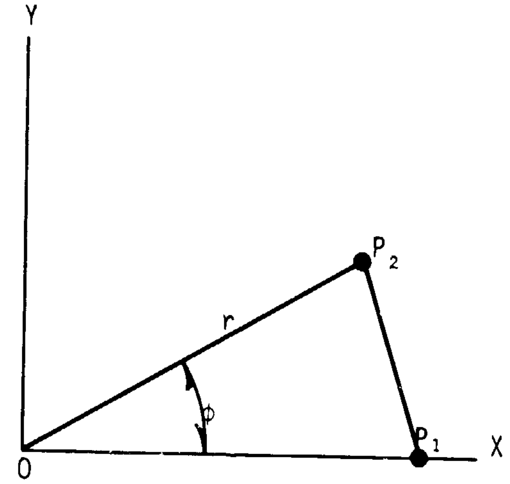


Figure 3

THEN: subdivide the angle ϕ into N angles θ

$$\theta = \frac{\phi}{N} \cdot T \qquad [3]$$

where T = Increment 1 to N

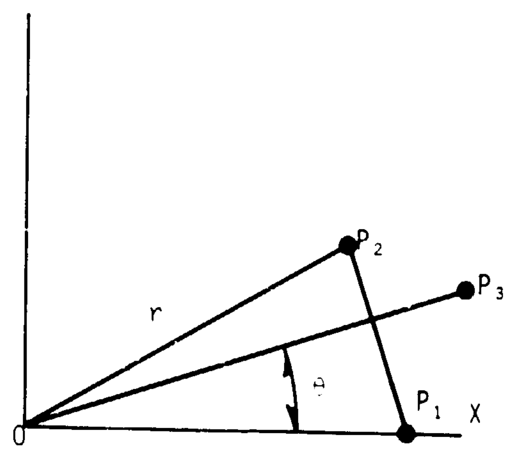


Figure 4

The points of intersection of $\overline{OP_3}$ and $\overline{P_1P_2}$ are found:

$$\overline{P_1P_2} \text{ is } \frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}$$

$$\overline{OP_3} \text{ is } \frac{y - 0}{x - 0} = \frac{y_3 - 0}{x_3 - 0} \qquad [4]$$

The equation takes the following form:

$$\overline{P_1P_2} \text{ is } x(y_2 - y_1) + y(x_1 - x_2) = y_1(x_1 - x_2) + x_1(y_2 - y_1)$$

let $(v_2 - y_1) = a_1$

$(x_2 - x_1) = b_1$

$y_1(x_1 - x_2) + x_1(y_2 - y_1) = C_1$

$$\overline{OP} \text{ is } xy_3 - yx_3 = 0$$

let $y_3 = a_2$

$-x_3 = b_2$

$0 = c_2$

Solve the equations for the point of intersection:

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} \qquad X = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{D} \qquad [5]$$

$$Y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{D}$$

Rotate the points of intersection along the PIT edge from 2-spaces back to 3-spaces.

$$\begin{aligned} x &= \lambda_1 x' + \mu_1 y' + \nu_1 z \\ y &= \lambda_2 x' + \mu_2 y' + \nu_2 z \\ z &= \lambda_3 x' + \mu_3 y' + \nu_3 z \end{aligned} \qquad [6]$$

where λ, μ, ν, are direction cosines of the X'-axis, Y'-axis, and Z'-axis with respect to the old axis and are found:

$$\lambda_1 = x_1 / \sqrt{x_1^2 + y_1^2 + z_1^2}$$

$$\mu_1 = y_1 / \sqrt{x_1^2 + y_1^2 + z_1^2}$$

$$\nu_1 = z_1 / \sqrt{x_1^2 + y_1^2 + z_1^2}$$

λ_2, λ_3; μ_2, μ_3; and ν_2, ν_3 are found similarly.

Retain the co-ordinates along the edges S_1, S_2 and S_3 as shown in Figure 5.

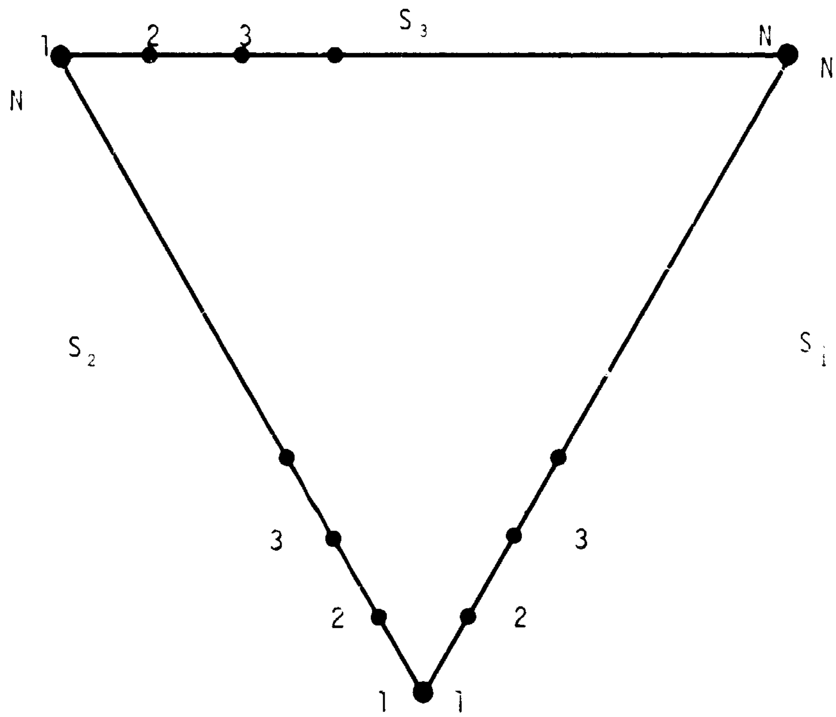


Figure 5

After finding the unit measurements along the edges of the PIT, they are connected through a 3-way grid determining a smaller triangular grid network. Since the units along the PIT edge are not equal, the 3-way gridding will create "windows". The centers of these "windows" must be found to establish the final 3-way grid network, Figure 6.

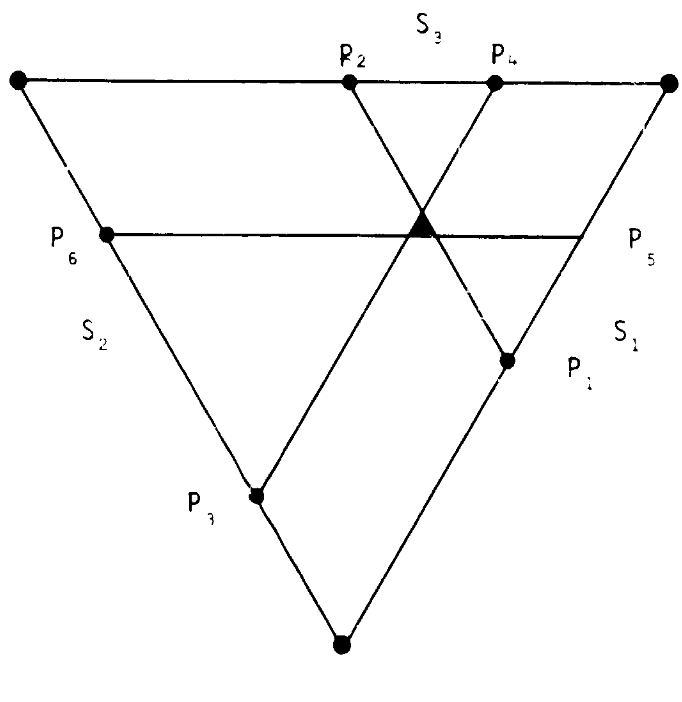


Figure 6

The gridding and windows are found by the following method: From the coordinates along the edges of S_1, S_2 and S_3, calculate coordinates of the window by finding the intersection of $\overline{P_1 P_2}$ with $\overline{P_3 P_4}$ and $\overline{P_1 P_2}$ with $\overline{P_5 P_6}$ and $\overline{P_3 P_4}$ with $\overline{P_5 P_6}$ by using the two point form of the equation of a line in three-space for the three lines and solve simultaneously for the points of intersection.

$$\overline{P_1P_2} \text{ is: } \quad \frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} = \frac{z - z_1}{z_2 - z_1}$$

$$\overline{P_3P_4} \text{ is: } \quad \frac{x - x_3}{x_4 - x_3} = \frac{y - y_3}{y_4 - y_3} = \frac{z - z_3}{z_4 - z_3} \qquad [7]$$

$$\overline{P_5P_6} \text{ is: } \quad \frac{x - x_5}{x_6 - x_5} = \frac{y - y_5}{y_6 - y_5} = \frac{z - z_5}{z_6 - z_5}$$

To find the intersection of $\overline{P_1P_2}$ with $\overline{P_3P_4}$ the equation takes the following form:

(1) $\overline{P_1P_2}$ is: $x(y_2-y_1) + y(x_1-x_2) = y_1(x_1-x_2) + x_1(y_2-y_1)$

(2) $\overline{P_1P_2}$ is: $y(z_2-z_1) + z(y_2-y_1) = z_1(y_1-y_2) + y_1(z_2-z_1)$

(3) $\overline{P_3P_4}$ is: $x(y_4-y_3) + y(x_3-x_4) = y_1(x_3-x_4) + x_1(y_4-y_3)$

(4) $\overline{P_3P_4}$ is: $y(z_4-z_3) + z(y_3-y_4) = z_3(y_3-y_4) + y_3(z_4-z_3)$

For $\overline{P_1P_2}$ let: $(y_2-y_1) = a_1$

$(x_1-x_2) = b_1$

$y_1(x_1-x_2) + x_1(y_2-y_1) = c_1$

For $\overline{P_3P_4}$ let: $(y_4-y_3) = a_2$

$(x_3-x_4) = b_2$

$y_1(x_3-x_4) + x_1(y_4-y_3) = c_2$

using the formula [6] solve for x and y coordinates of the intersections of $\overline{P_1P_2}$ with $\overline{P_3P_4}$.

Find the z coordinate:

For $\overline{P_1P_2}$ let: $(z_2-z_1) = a_1$

$(y_1-y_2) = b_1$

$z_1(y_1-y_2) + y_1(z_2-z_3) = c_1$

For $\overline{P_3P_4}$ let: $(z_4-z_3) = a_2$

$(y_3-y_4) = b_2$

$z_3(y_3-y_4) + y_3(z_4-z_3) = c_2$

The other two vertices of the window are found in a similar manner. Once the coordinates for the vertices of the window are determined, its center is found by one of the following two methods:

METHOD I:

On the Principle Icosahedral Triangular Plane the windows appear as equilateral triangles with vertices $P_1(x_1y_1z_1)$, $P_2(x_2y_2z_2)$, $P_3(x_3y_3z_3)$ as shown in Figure 7.

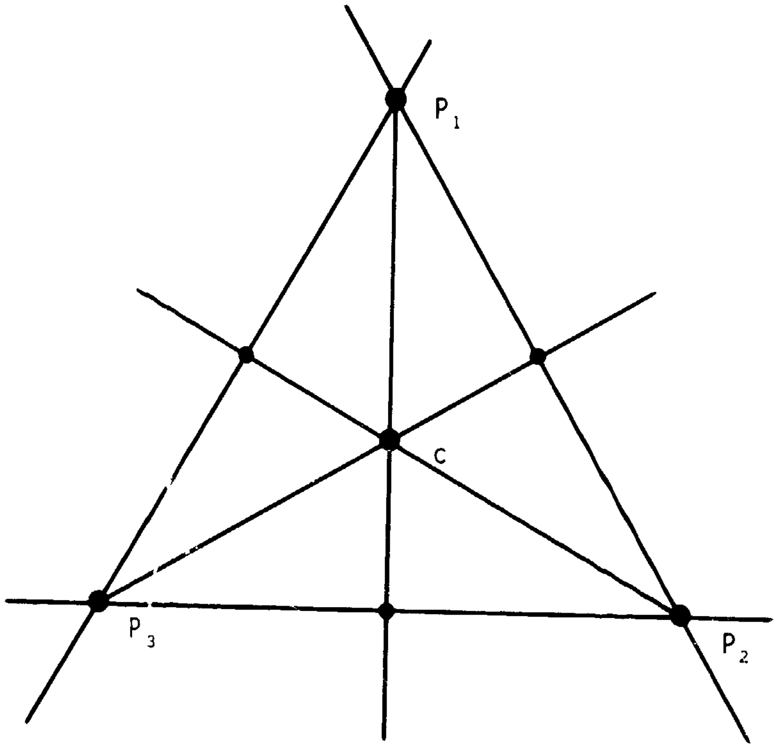


Figure 7

The center C(cx,cy,cz) is found with the following formula:

$$CX = \frac{x_1+x_2+x_3}{3}$$

$$CY = \frac{y_1+y_2+y_3}{3} \qquad [8]$$

$$CZ = \frac{z_1+z_2+z_3}{3}$$

METHOD II:

The coordinates of the window found on the surface of the PIT are first "exploded" to the surface of the sphere. The center of the exploded window is then found by the intersection of angle bisectors.

To find the projection of each vertex of the window onto the unit sphere, translate each vertex along a line through the vertex of the PIT and the origin; each coordinate of each vertex, PIT, is divided by the distance between the vertex PIT and the origin, Figure 8.

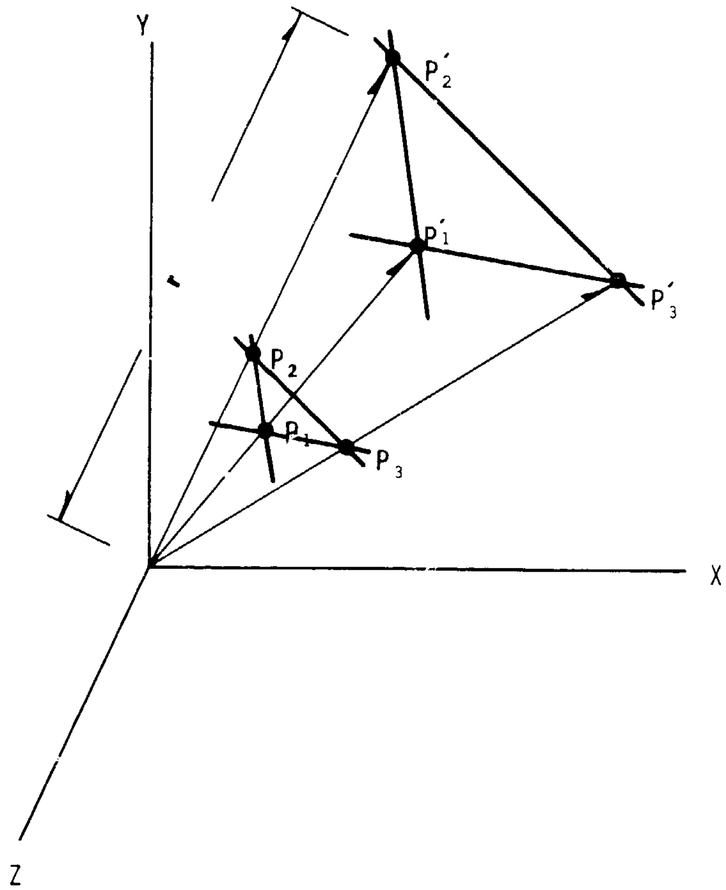


Figure 8

$$d = \sqrt{x_1^2 + y_1^2 + z_1^2} \quad [9]$$

where d = distance from origin to P_1

$$r = 1$$

where r = the radius of the sphere to be exploded upon and is considered constant

$$x'_1 = \frac{rx_1}{d}$$

$$y'_1 = \frac{ry_1}{d} \qquad [10]$$

$$z'_1 = \frac{rz_1}{d}$$

Translate "window" with vertice P_3 at the origin, Figure 9.

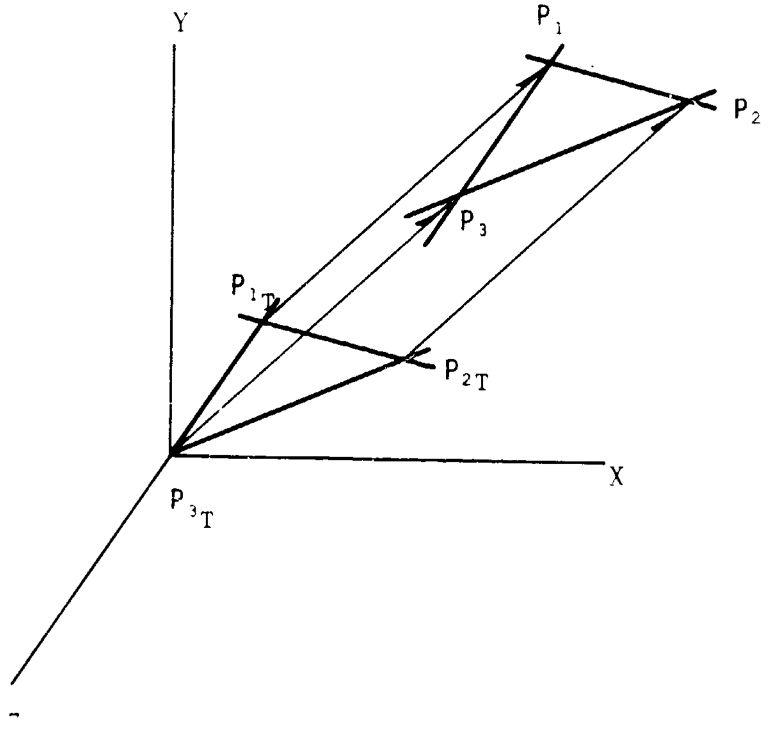


Figure 9

$$
\begin{aligned}
P_{1Tx} &= P_{1x} - P_{3x} \\
P_{1Ty} &= P_{1y} - P_{3y} \qquad\qquad [11] \\
P_{1Tz} &= P_{1z} - P_{3z} \\
\\
P_{2Tx} &= P_{2x} - P_{3x} \\
P_{2Ty} &= P_{2x} - P_{3x} \\
P_{2Tz} &= P_{2z} - P_{3z} \\
P_{3Tx_1}, P_{3Ty_1}, P_{3Tz_1} &= 0
\end{aligned}
$$

Rotate plane P_1, P_2, P_3 so that $\overline{P_1P_3}$ will fall on the X-axis and P_3 is at the origin using equation [1].

The center is found with the intersection of two angle bisectors of the triangular window $P_1P_2P_3$, Figure 10.

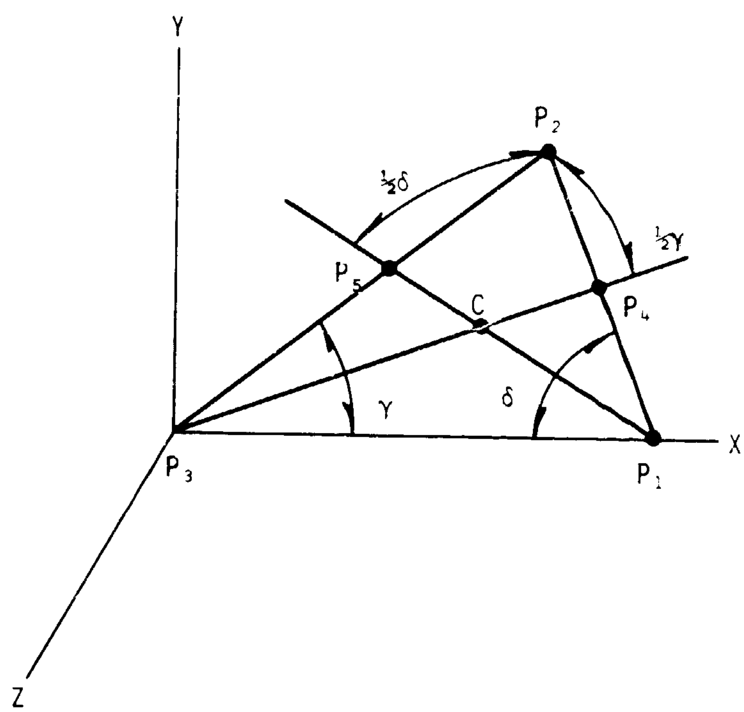


Figure 10

The angles γ and δ are found:

$$\text{Arctan} \quad \frac{y_2}{x_2} = \gamma$$

$$\text{Arctan} \quad \frac{y_2}{x_1 - x_2} = \delta \qquad [12]$$

rotate P_2 about P_3 toward P_1, $\frac{1}{2}\gamma$ degrees

$$x_4 = x_2 \cos \tfrac{1}{2}\gamma + y_2 \sin \tfrac{1}{2}\gamma$$
$$y_4 = y_2 \cos \tfrac{1}{2}\gamma - x_2 \sin \tfrac{1}{2}\gamma \qquad [13]$$

locate the origin at P_1, then rotate P_2 about P_1 toward P_3 $\frac{1}{2}\delta$ degrees.

$$x_5 = (x_2 - x_1) \cos \tfrac{1}{2}\delta - y_2 \sin \tfrac{1}{2}\delta + x_1$$
$$y_5 = y_2 \cos \tfrac{1}{2}\delta + (x_2 - x_1) \sin \tfrac{1}{2}\delta \qquad [14]$$

thus defining $\overline{P_1P_5}$ and $\overline{P_3P_4}$.

With P_3 at the origin formula [4] may be used to solve for the intersection of line $\overline{P_3P_4}$, $\overline{P_1P_5}$ finding center **C**. Rotate C back to three space using formula [6]. Then translate center C back to three space ("C" is located in the previously "exploded" window), Figure 11.

$$C'x = Cx + P_3x$$
$$C'y = Cy + P_3y \qquad [15]$$
$$C'z = Cz + P_3z$$

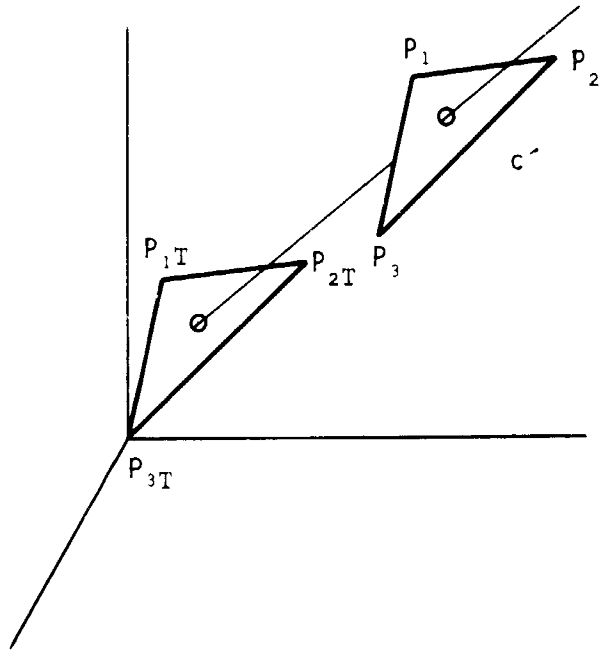


Figure 11

For Method I or Method II, the centers found are "exploded" to the surface of the sphere using formula [9] and formula [10].

Using the coordinates, this program finds the lengths of the elements of the structure (ℓ), the angle between pairs of elements (face angle α), the angle between the elements and a radius from the origin to an endpoint of the element (axial angle Ω), and the angle between adjacent faces of the structure (dihedral angle β), Figure 12.

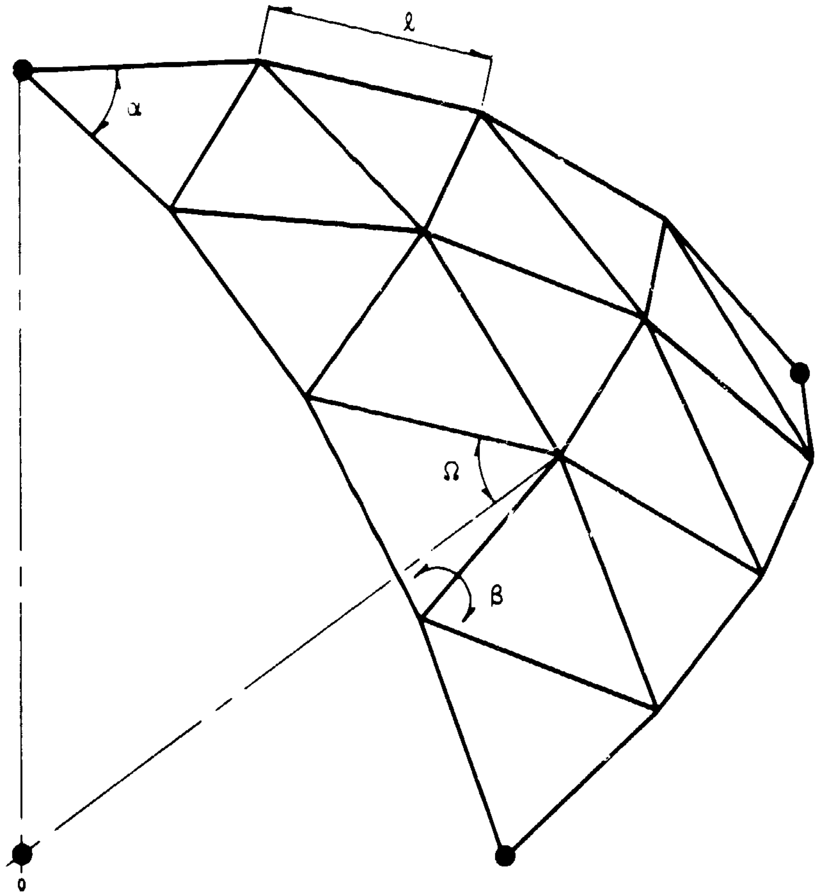


Figure 12

To find the angle between elements the face ∢ α, we use the coordinates of their endpoints. The vertex of the angle is a common endpoint to each element and is translated to the origin. The other two endpoints P_1 and P_2 are translated in the same manner. Letting (x_1, y_1, z_1) and (x_2, y_2, z_2) be the points resulting from the translations of the endpoints P_1 and P_2,

$$\cos \alpha = \left| \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{d_1 d_2} \right|$$

$$\text{where } d_1 = \sqrt{x_1^{\,2} + y_1^{\,2} + z_1^{\,2}}$$

$$\text{and } d_2 = \sqrt{x_2^{\,2} + y_2^{\,2} + z_2^{\,2}}$$

α is the desired angle.

To find axial angles the above method is used except that the vertex is established at one end of an element and the origin is used with the other endpoint to define the angle. The desired angle is Ω.

The angle between two adjacent faces, the dihedral $\measuredangle\ \beta$, is found using

$$\cos \beta = \frac{-\left|A_1 A_2 + B_1 B_2 + C_1 C_2\right|}{\sqrt{A_1^{\,2} + B_1^{\,2} + C_1^{\,2}}\ \sqrt{A_2^{\,2} + B_2^{\,2} + C_2^{\,2}}}$$

where

β is the desired angle.

$A_1X + B_1Y + C_1Z + D_1 = 0$ defines the plane cortaining one face and

$A_2X + B_2Y + C_2Z + D_2 = 0$ defines the plane containing the other face.

The negative sign is used because the obtuse angle is desired.

The A, B, and C for each plane are computed as

$$A = \begin{vmatrix} Y_1 & Z_1 & 1 \\ Y_2 & Z_2 & 1 \\ Y_3 & Z_3 & 1 \end{vmatrix}$$

$$B = \begin{vmatrix} X_1 & Z_1 & 1 \\ X_2 & Z_2 & 1 \\ X_3 & Z_3 & 1 \end{vmatrix}$$

$$C = \begin{vmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ X_3 & Y_3 & 1 \end{vmatrix}$$

where (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), and (X_3, Y_3, Z_3) lie in the plane. In particular the three vertices of each face are used.

The length of the elements ℓ are found by using the general equation:

$$\ell = \sqrt{(P_{x_1} - P_{x_2})^2 + (P_{y_1} - P_{y_2})^2 + (P_{z_1} - P_{z_2})^2}$$

ℓ is the desired length.

To reduce total output, this program takes into account certain symmetries and outputs only a part of the total angles and lengths. The rest of the values are the same as at least one outputed value and can easily be found using the following symmetries, Figure 13.

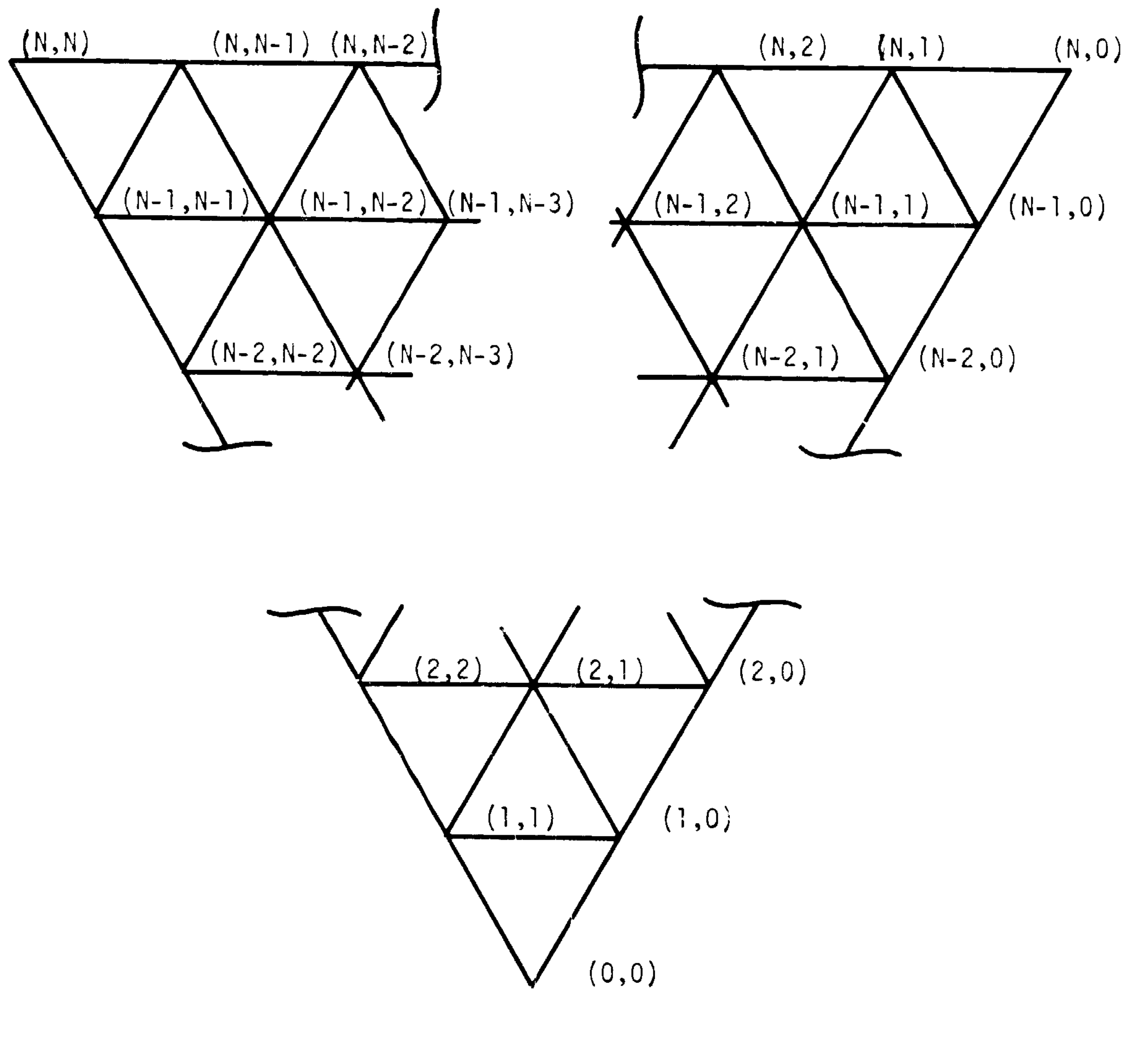

(N,N)
(N,N-1)
(N,N-2)
(N-1,N-1)
(N-1,N-2)
(N-1,N-3)
(N-2,N-2)
(N-2,N-3)
(N,2)
(N,1)
(N,0)
(N-1,2)
(N-1,1)
(N-1,0)
(N-2,1)
(N-2,0)
(2,2)
(2,1)
(2,0)
(1,1)
(1,0)
(0,0)

Figure 13

FACE ANGLES

For every face angle opening directly towards (or away from) the point (0,0), there are equal angles opening towards (or away from) the point (N,0) and (N,N). For example, the angle (1,1), (0,0), (1,0) with vertex at (0,0) is equal to the angle (N-1,0), (N,0), (N,1) and the angle (N,N-1), (N,N), (N-1, N-1). Thus, only the face angles facing directly towards or away from (0,0) are computed. If the vertex is to lie at (I,J), the angle will be either (I + 1, J + 1), (I, J), (I + 1, J) or (I-1, J-1), (I,J), (I-1, J). Also, only the face angles falling on the right of or on a line passing through (X_1, Y_1, Z_1,) and the midpoint of the opposite side are computed.

The elements of the structure can be put into one-to-one correspondence with the lengths and dihedral angles. The dihedral angle associated with an element is the angle between the two faces containing the element. For each element, there are two axial angles, one at each end, but since the element is a cord of the circle, the two angles are equal and may be considered one. In this case, we have a one-to-one correspondence between elements and axial angles. This program will only compute values around elements parallel to the side opposite (X_1, Y_1, Z_1) and on the right side of a line through (X_1, Y_1, Z_1) and the midpoint of the opposite side. All other lengths and angles are symetric to one of the lengths and angles computed in this manner.

PROJECTED RESEARCH

Another method is being developed of subdividing a polyhedron and "Exploding" it into a spherical form. A correlational study will be conducted to determine the difference in material usage, volume, and surface area of the subdivided forms, using the method described in this report, previously developed methods, methods presently under development, and the Fuller methods of subdividing a sphere.

A mathmatical model is also being developed which will explain the transformation of polyhedral forms as expandable structures.

APPENDIX I

The computer program here contained, was written for the IBM 7044 computer, utilizing FORTRAN language. The program may be used for a Tetrahedron, Octahedron, or Icosahedron, depending upon the coordinates used as input data. The output is given in units based upon a radius of 1 for the spherical form and therefore may be used as a basis for determining large structures.

The example of the output data given here is for a six frequency icosahedral sphere, and may be read as example I. The output takes advantage of symmetries within the spherical icosahedron as discussed in the text material. See Figure 14.

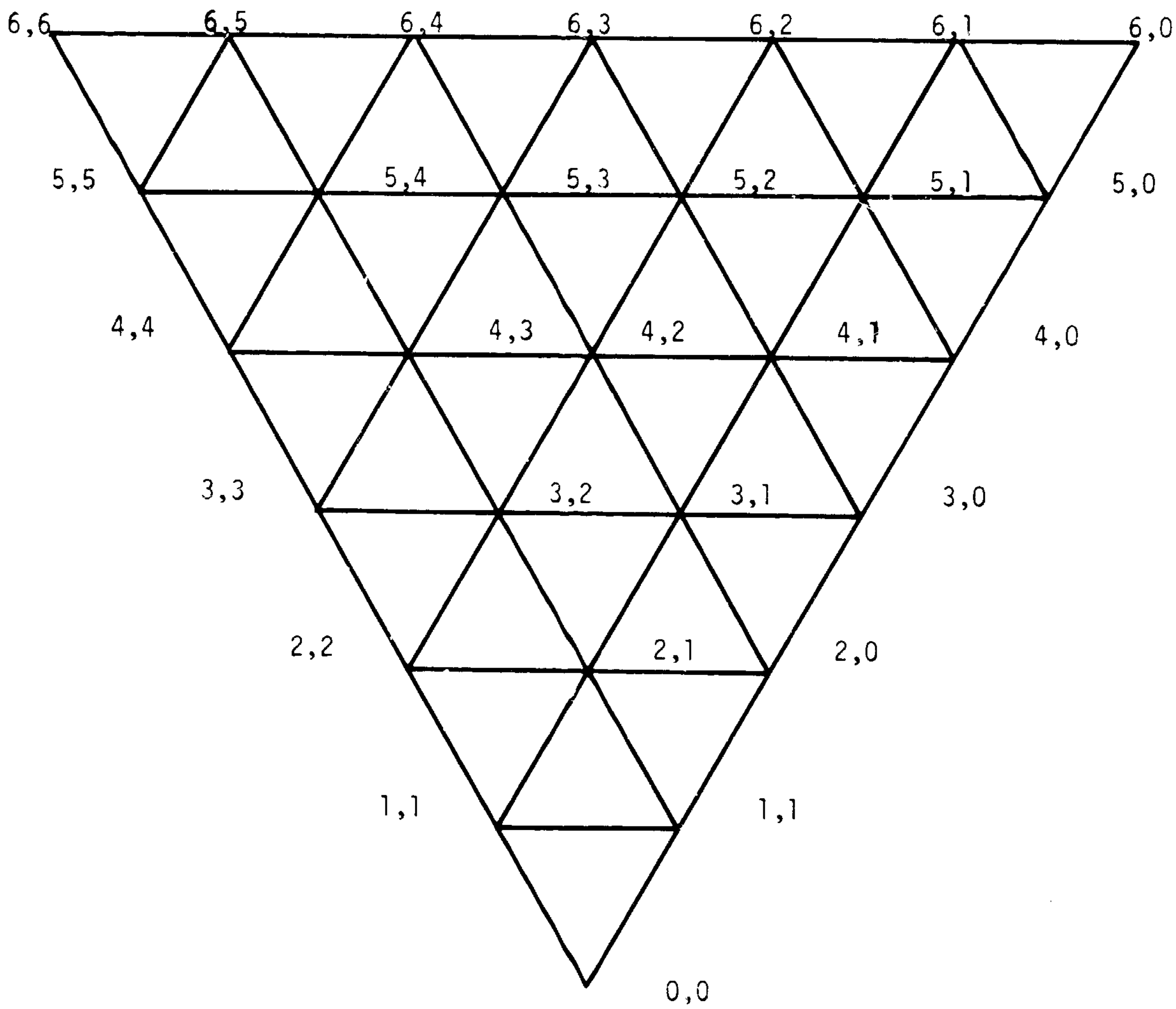

6,6
6,5
6,4
6,3
6,2
6,1
6,0
5,5
5,4
5,3
5,2
5,1
5,0
4,4
4,3
4,2
4,1
4,0
3,3
3,2
3,1
3,0
2,2
2,1
2,0
1,1
1,1
0,0

Figure 14

CHART G1

B1 Read the parameters for a DO loop governing the frequencies and two variables; one indicating the basic structure (tetrahedron, octahedron, or icosahedron) and another indicating what method of division will be used.

C1-C2 If the card was blank (first parameter zero), stop.

D1 Determine from the card read, what coordinates are to be used as the basic structure and the coefficients of some equations. This information is stored in an array by a DATA statement.

E1 Enter a DO loop governing the frequency.

F1 Set the summations for total area, length, and volume to zero. There are two summations of length: one for total structure and one above a face of the basic structure.

G1 Write a leader record containing the name of the basic structure, the frequency, and the method of division.

H1-J1 Find and write the total number of faces, edges, and vertices both in the entire structure and above one face of the basic structure. These figures are used for display only and are not used in any further calculations.

K1 Enter DO loop to determine the first end point of the edge.

A3 Enter DO loop to determine the second end point of the edge.

B3 The end points of the edges should not be the same.

C3 Rotate the plane defined by the end points and the origin into the X-Y plane with one of the end points into the X axis, and calculate the angle ϕ = ARCTAN (y^2/x^2) necessary for subdivision.

D3 Enter DO loop to determine the subdivision of the edge.

E3 Generate points of subdivision along the edge by N subdivisions of angle ϕ.

F3 Rotate the generated points back into the original plane.

G3 Store the points generated into their respective arrangement.

H3-K3-J3 If DO loop is complete there should be no more points on their respective edges.

A4 The three vertices of the basic structure are arbitrarily ordered. The first vertice is exploded to the surface of the unit sphere to explode a point. This is done by dividing the coordinates of a point by its distance from the origin.

B4 Print out the coordinate of this point: G2J1-G2K1; G2E2-G2F2; G2C3-G2D3.

C4 G1J5-G1K5; G2F1-G2G1; these blocks are used to count lines printed and to provide margins at the top and bottom of each page.

D4 Subdivision of the basic face will be generated by using the coordinates stored in SD1, SD2, and SD3 and taking all lines parallel to their respective edges. The intersections of these lines creates windows of which their coordinates are necessary for finding their centers (two methods of finding the windows center are used) for completion of the subdivision. Enter DO loop which generates points along the edge from the first vertex to the second.

E3 Enter a DO loop which generates points along the edge from the first vertex to the second. These points will be the first in each row of points parallel to the edge from the second vertex to the third. The first row will be the one nearest the first vertex.

E4 Enter DO loop which generates the points along the Tow determined in E4.

F4 Determine the I. J. values of the basic unit.

G4 Enter DO loop which generates the coordinates of the basic unit.

H4 Set up constants of points in basic units used in determining coordinates of the windows.

J4-K4 All points should be within the PIT. If not, set equal to zero (necessary for finding dihedral angles along PIT edges.)

A5-B5 Points along the PIT edges may be found directly from an SD array.

C5 INTERC is a subroutine that finds the coordinates of the intersection of 2 lines; used to find the coordinates of the vertice of the window.

D5 Determine the center of the window by one of two methods: Method #1: find the center by taking the intersection of four lines, each passing thru a vertice of the window and the midpoints of the opposite side.

E5 This is the end of the DO loop that finds all of the points within the basic unit.

F5 This explodes all points of the basic unit to the surface of the sphere.

H5 Writes the coordinates of P_1 and P_3

CHART GII

B1 Due to symmetry, not all angles and lengths are needed.

C1 Find the length between P_1 and P_2.

D1 Add a multiple of this length into the sums. A multiple is used because other elements symmetric to it will not be found due to the conditions of B1.

E1 Lable and write this length.

H1 Find the axial angle with vertex angle at P_1 and legs through P_3 and the origin.

J1 Lable and write this axial angle.

B2 Find the total area of and volume under the face generated by P_1, P_2, and P_3.

C2 Add this into the summations.

D2 Find the face angles with vertices at P_2 and P_4 and legs passing through P_1 and P_3.

E2 Lable and write the face angles.

H2-K2 If P_1 is in the last row of the face of the basic structure, then there is not face between P_1, P_3, and P_4. Otherwise, the area of and volume under that face will be added into the summations.

A3 Find the dihedral angle between the two faces formed by P_1, P_2, P_3, and P_4. If P_4 is not above this face of the basic structure, it will not be properly generated. In this case, find the angle between the face formed by P_1, P_2, and P_3, and a plane passing through P_1, P_3, and the origin, doubling this result.

B3 Lable and write this dihedral angle.

E3 This is the end of the DO loop of block GIE4. If the loop hasn't been satisfied, the next point will be the first in the next row.

G3 This is the end of DO loop of block GID4. If the loop hasn't been satisfied, the next point will be the first in the next row.

H3-J3 Find, lable, and write the total areas, volumes, and lengths of the structure.

K3 This is the end of the DO loop of block GIE1. If the loop hasn't been satisfied, the same basic structure will be used for the next frequency. Otherwise a new card is read.

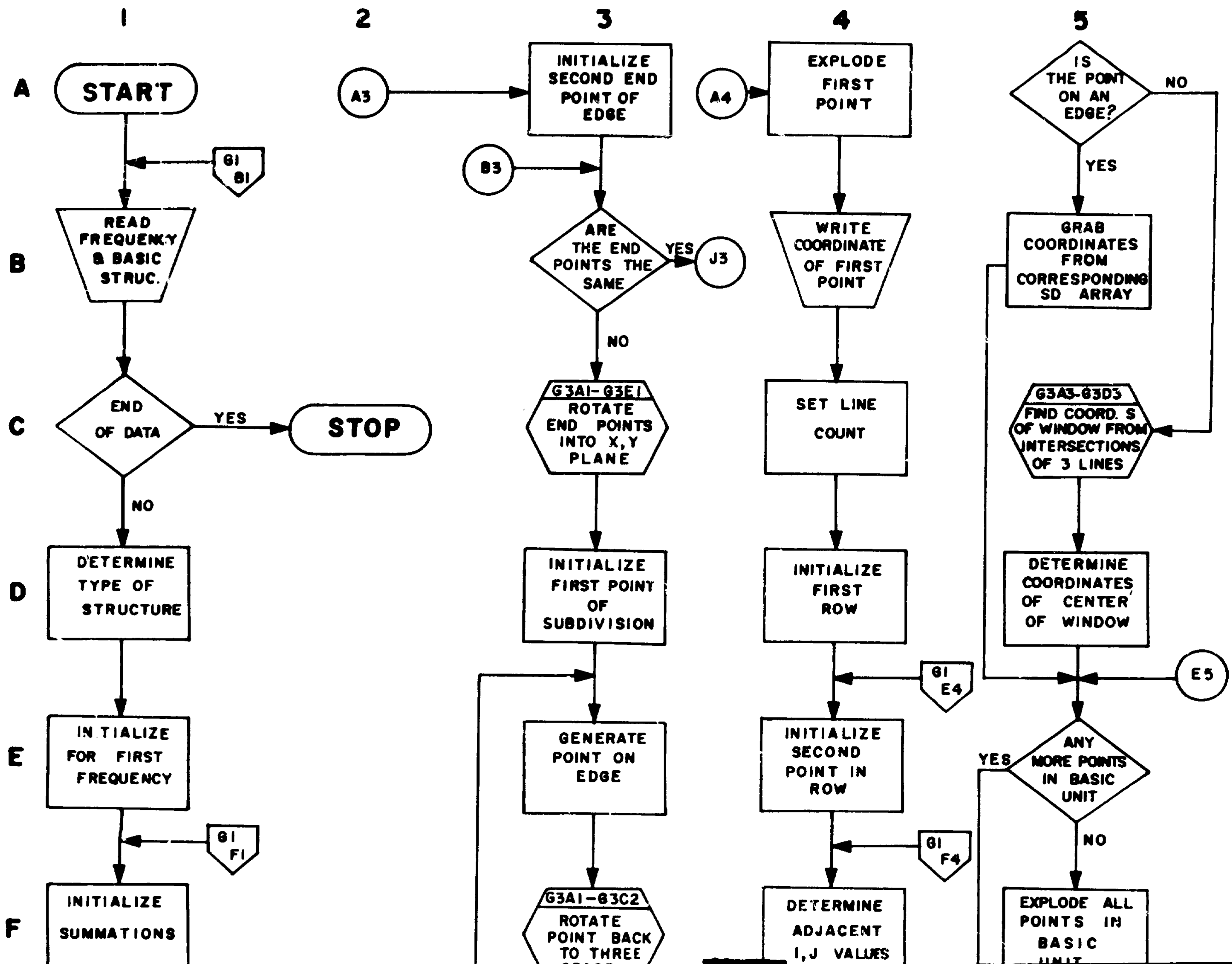
As used in blocks G1F1, G1H1-G1J1:

the vertices above a face are those vertices exploded from points on that face. The lengths and areas above the face are those created by the vertices above the face.

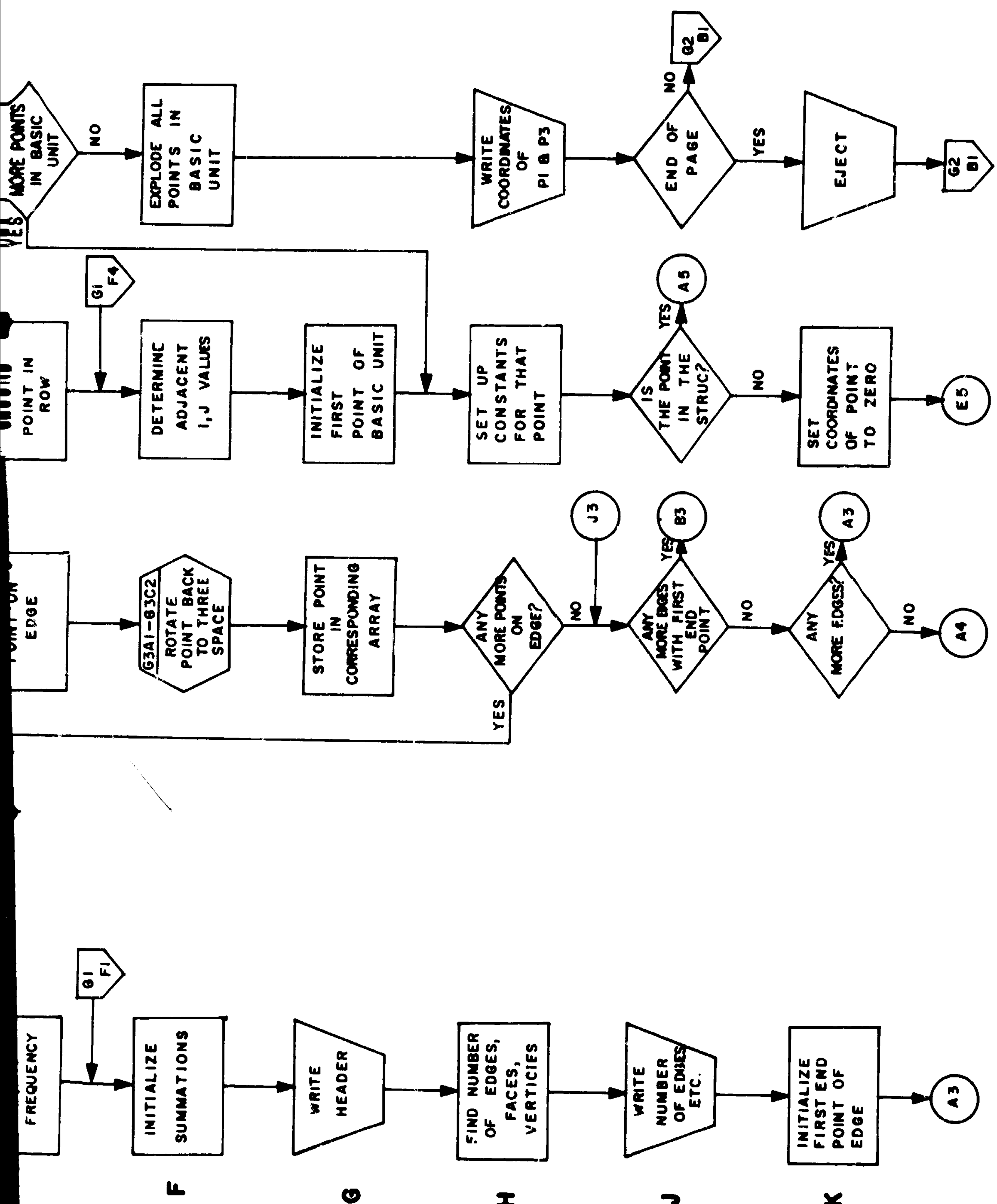
As used in blocks G1H1-G1J1, G1E5:

the basic unit consists of four points combined in two adjacent triangles with two parts approximately parallel to the edge containing vertex 2 and 3.

Chart ID: G1 Chart Name: GEOD23

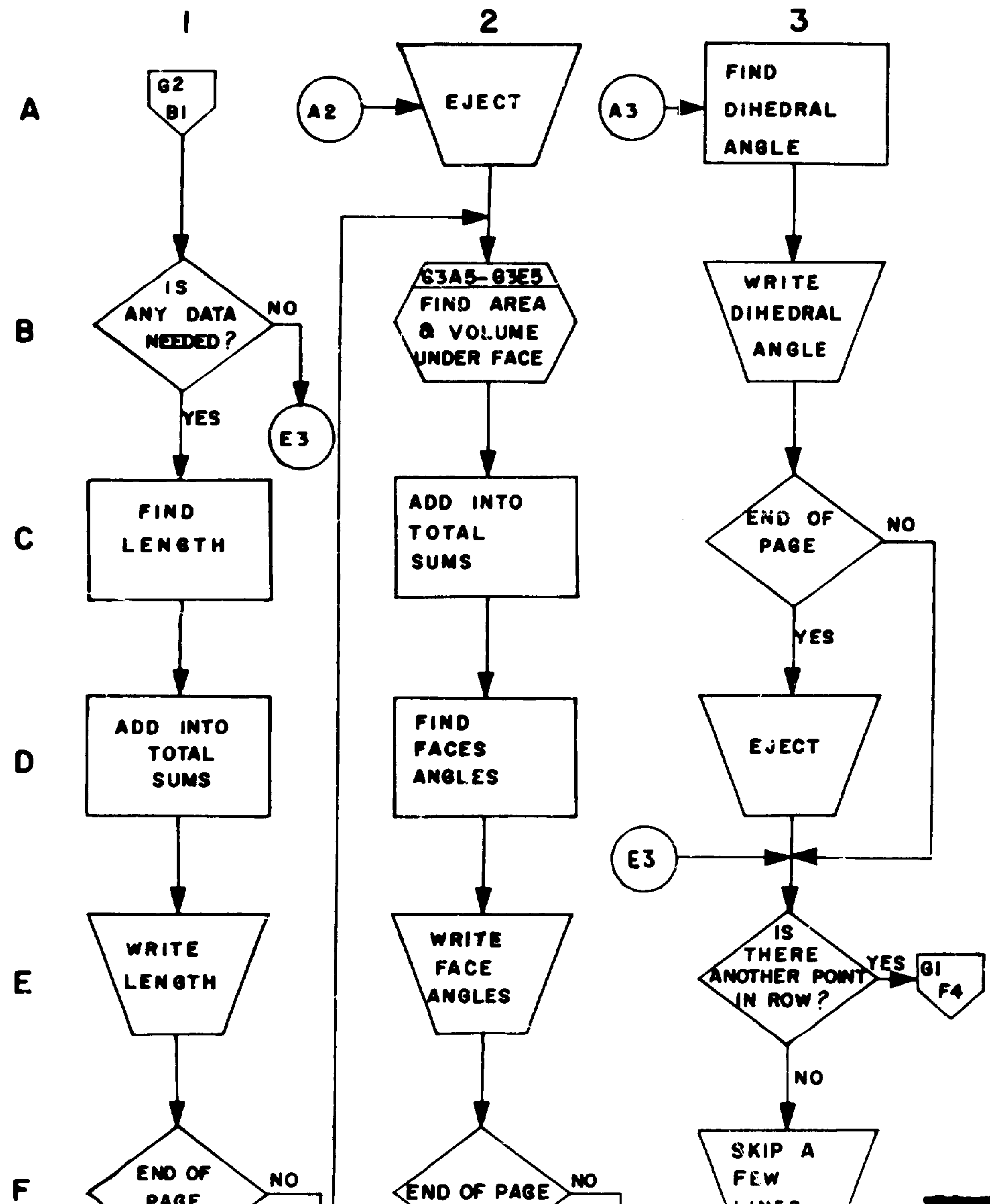


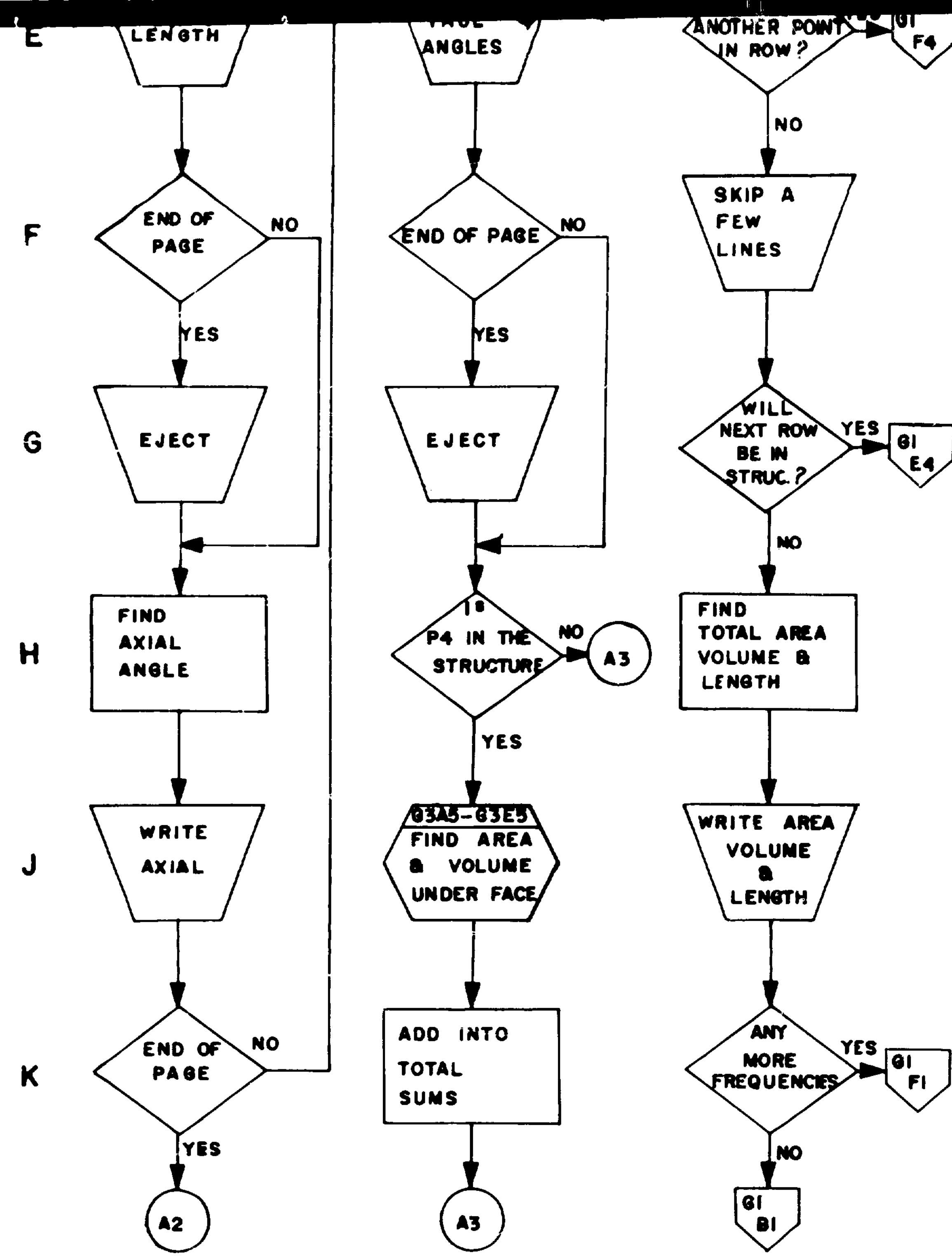
FOLDOUT FRAME 1



FOLDOUT FRAME 2

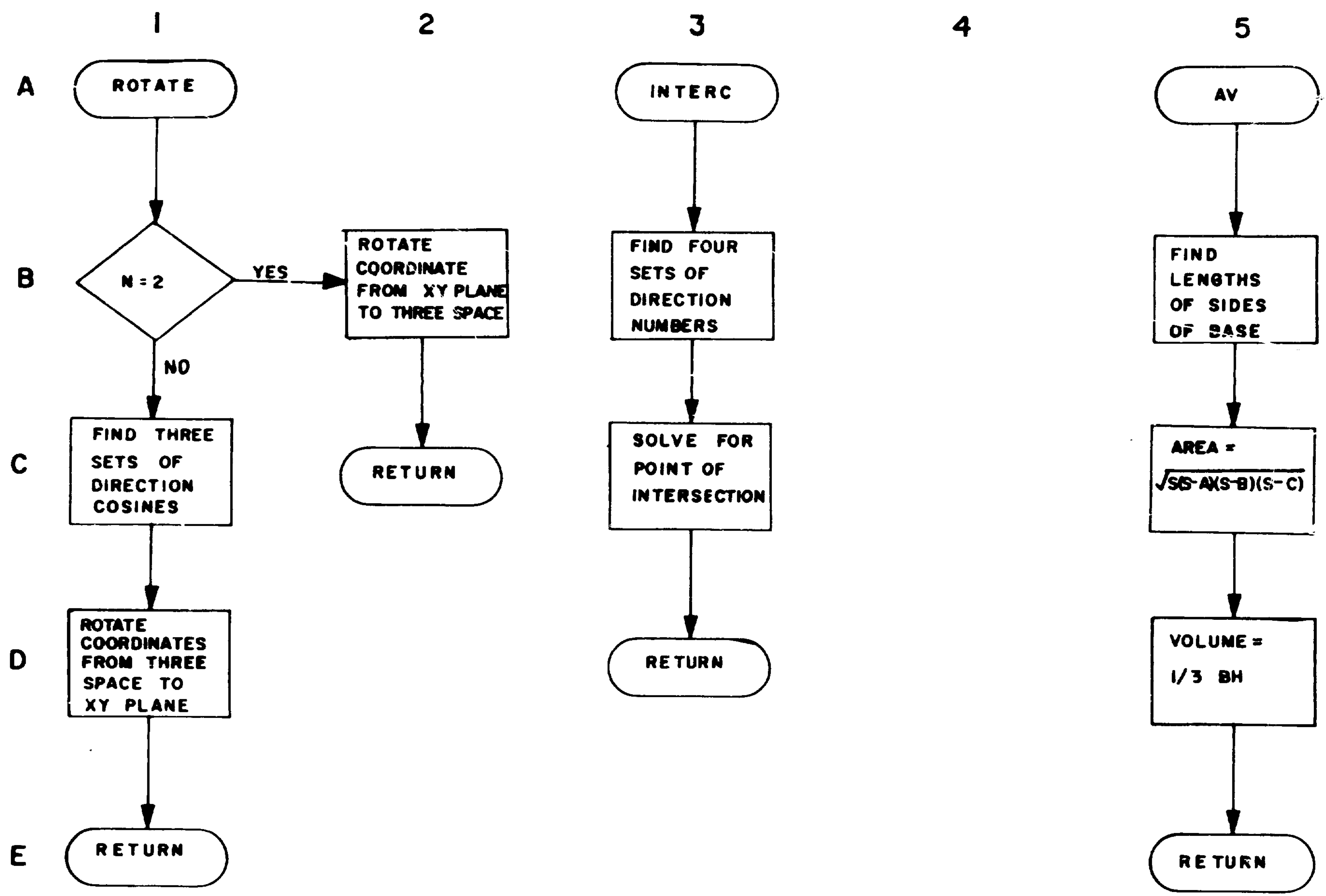
Chart ID: G2 Chart Name: GEOD 23





FOLDOUT FRAME 2

Chart ID: G3 Chart Name GEOD 23



```
$IBFTC GEOD    NODECK
      DIMENSION SD 1(3, 32), SD 2(3, 32), SD 3(3, 32) , P(4, 3)
      DIMENSION COORD(3, 3, 3), V(3, 3), LAB(2, 4), NAMES(2, 3),
     1  FIG(2, 3)
      INTEGER EFV(3, 3), PAG SIZ, TE OC IC, VL, EL, FL, VG, EG, FG
      REAL L OF TRI, LENGTH, L OF FIG
      DATA FIG / 4., 8.,  8., 16.,  20., 40. /
      DATA NAMES / 12H TETRAHEDRON, 12H OCTAHEDRON , 12H ICOSAHEDRON /,
     1  PAG SIZ / 56 /
      DATA COORD / -.57735027, -.57735027, .57735027, .57735027,
     1  -.57735027, -.57735027, -.57735027, .57735027, -.57735027,
     2  1., 0., 0., 0., 0., 1., 0., 1., 0.,  .0, .85065081, .52573111,
     3  .85065081, .52573111, 0., .52573111, 0., .85065081  /
      DATA EFV / 4, 8, 20, 2, 4, 10, 6, 12, 30 /
      EQUIVALENCE
     1  (V(1, 1), X1), (V(2, 1), Y1), (V(3, 1), Z1),
     2  (V(1, 2), X2), (V(2, 2), Y2), (V(3, 2), Z2),
     3  (V(1, 3), X3), (V(2, 3), Y3), (V(3, 3), Z3),
     4  (LAB(1, 1), I1), (LAB(2, 1), J1), (LAB(1, 2), I2),
     5  (LAB(2, 2), J2), (LAB(1, 3), I3), (LAB(2, 3), J3),
     6  (LAB(1, 4), I4), (LAB(2, 4), J4)
      COEF(A1, A2, A3, B1, B2, B3) = A1 * (B2 - B3) - A2 * (B1 - B3) +
     1  A3 * (B1 - B2)
   70 FORMAT (4I2)
   71 FORMAT (10H1FREQUENCY I5, 2X, 2A6, 5X, 8HMETHOD 2)
   72 FORMAT (7HOV(L) = I5, 9H   E(L) = I5, 9H   F(L) = I5, 9H   V(G) =
     *  I5, 9H   E(G) = I5, 9H   F(G) = I5  ///)
   73 FORMAT (9HL   0   0 3(5X, F10.6)///)
   74 FORMAT (1HL 2I4, 3(5X, F10.6))
   75 FORMAT (1HA)
   76 FORMAT (9X, 8HLENGTH  , 2(3X, 2I4), 21X, F11.8)
   77 FORMAT (9X, 8HAXIAL   , 11H   0.0 0.0, 2(3X, 2I4), 10X, 2F11.6)
   78 FORMAT (9X, 8HFACE    , 3(3X, 2I4), 10X, 2F11.6)
   79 FORMAT (9X, 8HDIHEDRAL, 2(3X, 2I4), 21X, 2F11.6)
   80 FORMAT (//)
   81 FORMAT (12H1   ONE FACE/ 9X, 4HAREA, 27X, F15.6, / 9X, 6HVOLUME,
     *  25X, F15.6 / 9X, 6HLENGTH, 25X, F15.6 // 22H   COMPLETE STRUCTU
     *RE / 9X, 4HAREA, 27X, F15.6 / 9X, 6HVOLUME, 25X, F15.6 / 9X,
     *  6HLENGTH, 25X, F15.6)
    1 READ (5, 70) MIN, MAX, INC, TE OC IC
      IF (MIN .EQ. 0) STOP
      IF (MAX .EQ. 0) MAX = MIN
      IF (INC .EQ. 0) INC = 1
      K = 1
      IF (TE OC IC .EQ. 8) K = 2
      IF (TE OC IC .EQ. 20) K = 3
      DO 2 I = 1, 3
      DO 2 J = 1, 3
    2 V(J, I) = COORD(J, I, K)
      DO 25 N = MIN, MAX, INC
      FLOAT N = N
      SUM OF L = 0.
      SUM OF A = 0.
      SUM OF V = 0.
      L OF TRI = 0.
```

```
      WRITE (6, 71) N, NAMES(1, K), NAMES(2, K)
      VL = (N + 1) * (N + 2) / 2
      EL = 3 * N * (N + 1) / 2
      FL = N**2
      VG = EFV (K, 2) * N**2 + 2
      EG = EFV (K, 3) * N**2
      FG = EFV (K, 1) * FL
      WRITE (6, 72) VL, EL, FL, VG, EG, FG
      L1 = 0
      DO 5 L2 = 1, 2
      DO 5 LN = 2, 3
      IF (L2 .EQ. LN) GO TC 5
      L1 = L1 + 1
      CALL ROTATE (1, V(1, LN), V(2, LN), V(3, LN), X2R, Y2R, Z2R,
     1  V(1, L2), V(2, L2), V(3, L2))
      THETA = ATAN2(Y2R, X2R)
      DO 5 L4 = 1, N
      T = L4
      ANG = T * THETA / FLCAT N
      X = COS(ANG)
      Y = SIN(ANG)
      A1 = -Y
      B1 = X
      A2 = -Y2R
      B2 = X2R - 1.
      C2 = B2
      Y = (A1*C2) / (A1*B2 - A2*B1)
      X = (B1*C2) /(A2*B1 - A1*B2)
      CALL RCTATE (2, X, Y, 0.,XR, YR, ZR)
      IF (L1 - 2) 27, 3, 4
   27 SD 1(1, L4) = XR
      SD 1(2, L4) = YR
      SD 1(3, L4) = ZR
      GO TO 5
    3 SD 2(1, L4) = XR
      SD 2(2, L4) = YR
      SD 2(3, L4) = ZR
      GO TO 5
    4 SD 3(1, L4) = XR
      SD 3(2, L4) = YR
      SD 3(3, L4) = ZR
    5 CONTINUE
      R = SQRT(X1**2 + Y1**2 + Z1**2)
      P(1, 1) = X1 / R
      P(1, 2) = Y1 / R
      P(1, 3) = Z1 / R
      WRITE (6, 73) P(1, 1), P(1, 2), P(1, 3)
      LINES = 12
      DO 24 I1 = 1, N
      DO 23 J1 = 1, I1
      I2 = I1 - 1
      J2 = J1 - 1
      I3 = I1
      J3 = J2
      I4 = I1 + 1
```

```
      J4 = J1
      DO 11  LA = 1, 4
      LI = LAB(1, LA)
      LJ = LAB(2, LA)
      IJ = LI - LJ
      NJ = N - LJ
      IF (LI .LE. N) GO TC 65
      P(LA, 1) = 0.
      P(LA, 2) = 0.
      P(LA, 3) = 0.
      GO TO 11
   65 IF (LI .EQ. LJ) GC TC 7
      IF (LJ .EQ. 0) GO TC  9
      IF (LI .NE. N) GO TC 10
      P(LA, 1) = SD 3(1, NJ)
      P(LA, 2) = SD 3(2, NJ)
      P(LA, 3) = SD 3(3, NJ)
      GO TO 11
    7 IF (LJ .EQ. 0) GO TC 8
      P(LA, 1) = SD 1(1, LI)
      P(LA, 2) = SD 1(2, LI)
      P(LA, 3) = SD 1(3, LI)
      GO TO 11
    8 P(LA, 1) = X1
      P(LA, 2) = Y1
      P(LA, 3) = Z1
      GO TO 11
    9 P(LA, 1) = SD 2(1, LI)
      P(LA, 2) = SD 2(2, LI)
      P(LA, 3) = SD 2(3, LI)
      GO TO 11
   10 CALL INTERC (SD 1(1, LI), SD 1(2, LI), SD 1(3, LI),SD2(1, LI),
     1  SD 2(2, LI), SD 2(3, LI), SD 3(1, NJ), SD 3(2, NJ), SD 3(3, NJ),
     2  SD 1(1, LJ), SD 1(2, LJ), SD 1(3, LJ), XE1, YE1, ZE1)
      CALL INTERC (SD 3(1, IJ), SD 3(2, IJ), SD 3(3, IJ), SD 2(1, IJ),
     1  SD 2(2, IJ), SD 2(3, IJ), SD 3(1, NJ), SD 3(2, NJ), SD 3(3, NJ),
     2  SD 1(1, LJ), SD 1(2, LJ), SD 1(3, LJ), XE2, YE2, ZE2)
      CALL INTERC (SD 3(1, IJ), SD 3(2, IJ), SD 3(3, IJ), SD 2(1, IJ),
     1  SD 2(2, IJ), SD 2(3, IJ), SD 1(1, LI), SD 1(2, LI), SD 1(3, LI),
     2  SD 2(1, LI), SD 2(2, LI), SD 2(3, LI), XE3, YE3, ZE3)
C     BEGIN DETERMINATICN CF CENTER OF WINDOW IN PLANE OF MAJOR FACE . . .
      P(LA, 1) = (XE1 + XE2 + XE3) / 3.
      P(LA, 2) = (YE1 + YE2 + YE3) / 3.
      P(LA, 3) = (ZE1 + ZE2 + ZE3) / 3.
   11 CONTINUE
      L = 4
      IF (I4 .GT. N) L = 3
      DO 12 INDEX 1 = 1, L
      R = SQRT(P(INDEX 1, 1)**2 + P(INDEX 1, 2)**2 + P(INDEX 1, 3)**2)
      DO 12 INDEX 2 = 1, 3
   12 P(INDEX 1, INDEX 2) = P(INDEX 1, INDEX 2) / R
      IF (J3 .NE. 0) GO TC 125
      WRITE (6, 74) I3, J3, P(3, 1), P(3, 2), P(3, 3)
      LINES = LINES + 3
      IF (LINES .LT. PAG SIZ) GC TO 125
```

```
      WRITE(6, 75)
      LINES = 1
  125 WRITE (6, 74) I1, J1, P(1, 1), P(1, 2), P(1, 3)
      LINES = LINES + 3
      IF (LINES .LT. PAG SIZ) GO TO 13
      WRITE (6, 75)
      LINES = 1
   13 IF (2 * J3 .GE. I3) GO TO 23
      LENGTH = SQRT((P(1, 1) - P(3, 1))**2 + (P(1, 2) - P(3, 2))**2 +
     *  (P(1, 3) - P(3, 3))**2)
      L OF TRI = L OF TRI + 3. * LENGTH
      IF (2 * J1 .GT. I1) GO TO 14
      L OF TRI = L OF TRI + 3. * LENGTH
      SUM OF L = SUM OF L + 1.5 * LENGTH
      IF (I1 .LT. N) SUM OF L = SUM OF L + 1.5 * LENGTH
      GO TO 15
   14 SUM OF L = SUM OF L + .75 * LENGTH
      IF (I1 .LT. N) SUM OF L = SUM OF L + .75 * LENGTH
   15 WRITE (6, 76) I1, J1, I3, J3, LENGTH
      LINES = LINES + 1
      IF (LINES .LT. PAG SIZ) GO TO 16
      WRITE (6, 75)
      LINES = 1
   16 XT = P(1, 1) - P(3, 1)
      YT = P(1, 2) - P(3, 2)
      ZT = P(1, 3) - P(3, 3)
      D1D2 = SQRT (XT**2 + YT**2 + ZT**2)
      ANG = ARCOS(ABS(XT*P(3, 1) + YT*P(3, 2) + ZT*P(3, 3))/ D1D2)
      DEG = ANG / .017453293
      WRITE (6, 77) I3, J3, I1, J1, DEG, ANG
      LINES = LINES + 1
      IF (LINES .LT. PAG SIZ) GO TO 17
      WRITE (6, 75)
      LINES = 1
   17 CALL AV(P(1, 1), P(1, 2), P(1, 3), P(3, 1), P(3, 2), P(3, 3),
     *  P(2, 1), P(2, 2), P(2, 3), AREA, VOLUME)
      IF (2 * J1 .LE. I1) GO TO 18
      AREA = .5 * AREA
      VOLUME = .5 * VOLUME
   18 SUM OF A = SUM OF A + AREA
      SUM OF V = SUM OF V + VOLUME
      DO 19 INDEX 1 = 2, 4, 2
      XA = P(1, 1) - P(INDEX 1, 1)
      YA = P(1, 2) - P(INDEX 1, 2)
      ZA = P(1, 3) - P(INDEX 1, 3)
      XB = P(3, 1) - P(INDEX 1, 1)
      YB = P(3, 2) - P(INDEX 1, 2)
      ZB = P(3, 3) - P(INDEX 1, 3)
      D1D2 = SQRT((XA**2 + YA**2 + ZA**2)) * SQRT((XB**2 + YB**2 +
     *  ZB**2))
      ANG = ARCOS((XA*XB + YA*YB + ZA*ZB) / D1D2)
      DEG = ANG / .017453293
      WRITE (6, 78) I1, J1, LAB(1, INDEX 1), LAB(2, INDEX 1), I3, J3,
     1  DEG, ANG
      LINES = LINES + 1
```

```
      IF (LINES .LT. PAG SIZ) GO TO 19
      WRITE (6, 75)
      LINES = 1
   19 IF (I4 .GT. N) GO TO 21
      CALL AV(P(1, 1), P(1, 2), P(1, 3), P(3, 1), P(3, 2), P(3, 3),
     *  P(4, 1), P(4, 2), P(4, 3), AREA, VOLUME)
      IF (2 * J1 .LE. I1) GO TO 20
      AREA = .5 * AREA
      VOLUME = .5 * VOLUME
   20 SUM OF A = SUM OF A + AREA
      SUM OF V = SUM OF V + VOLUME
   21 A1 = COEF(P(1, 2), P(2, 2), P(3, 2), P(1, 3), P(2, 3), P(3, 3))
      B1 = COEF(P(1, 1), P(2, 1), P(3, 1), P(1, 3), P(2, 3), P(3, 3))
      C1 = COEF(P(1, 1), P(2, 1), P(3, 1), P(1, 2), P(2, 2), P(3, 2))
      A2 = COEF(P(1, 2), P(4, 2), P(3, 2), P(1, 3), P(4, 3), P(3, 3))
      B2 = COEF(P(1, 1), P(4, 1), P(3, 1), P(1, 3), P(4, 3), P(3, 3))
      C2 = COEF(P(1, 1), P(4, 1), P(3, 1), P(1, 2), P(4, 2), P(3, 2))
      D1D2 = SQRT((A1**2 + B1**2 + C1**2) * (A2**2 + B2**2 + C2 **2))
      ANG = ARCOS(-ABS(A1*A2 + B1*B2 + C1*C2) / D1D2)
      IF (I4 .GT. N) ANG = 2. * (3.1415927 - ANG)
      DEG = ANG / .017453293
      WRITE (6, 79) I1, J1, I3, J3, DEG, ANG
      LINES = LINES + 1
      IF (LINES .LT. PAG SIZ) GO TO 23
      WRITE (6, 75)
      LINES = 1
   23 CONTINUE
      WRITE (6, 80)
      LINES = LINES + 3
      IF (LINES .LT. PAG SIZ) GO TO 24
      WRITE (6, 75)
      LINES = 1
   24 CONTINUE
      A OF TRI = 2. * SUM OF A
      V OF TRI = 2. * SUM OF V
      A OF FIG = FIG(1, K) * A OF TRI
      V OF FIG = FIG(1, K) * V OF TRI
      L OF FIG = FIG(2, K) * SUM OF L
      WRITE (6, 81) A OF TRI, V OF TRI, L OF TRI, A OF FIG, V OF FIG,
     *  L OF FIG
   25 CONTINUE
      GO TO 1
      END
```

```
$IBFTC ROTATE  NODECK
      SUBROUTINE ROTATE (N, X2, Y2, Z2, X2R, Y2R, Z2R, X1, Y1, Z1)
      IF (N .EQ. 2) GO TO 1
      D1 = SQRT(X1**2 + Y1**2 + Z1**2)
      H1 = X1 / D1
      U1 = Y1 / D1
      V1 = Z1 / D1
      H3 = Y1 * Z2 - Y2 * Z1
      U3 = Z1 * X2 - X1 * Z2
      V3 = X1 * Y2 - X2 * Y1
      D2= SQRT(H3**2 + U3**2 + V3**2)
      H3 = H3 / D2
      U3 = U3 / D2
      V3 = V3 / D2
      H2 = U3 * Z1 - Y1 * V3
      U2 = V3 * X1 - Z1 * H3
      V2 = H3 * Y1 - X1 * U3
      D3 = SQRT(H2**2 + U2**2 + V2**2)
      H2 = H2 / D3
      U2 = U2 / D3
      V2 = V2 / D3
      X2R = X2 * H1 + Y2 * U1 + Z2 * V1
      Y2R = X2 * H2 + Y2 * U2 + Z2 * V2
      RETURN
    1 X2R = X2 * H1 + Y2 * H2 + Z2 * H3
      Y2R = X2 * U1 + Y2 * U2 + Z2 * U3
      Z2R = X2 * V1 + Y2 * V2 + Z2 * V3
      RETURN
      END
```

```
$IBFTC INTERC  NODECK
      SUBROUTINE INTERC (X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, X4, Y4, Z4,
     *  X, Y, Z)
      A1 = Y2 - Y1
      B1 = X1 - X2
      C1 = X1*(Y1 - Y2) + Y1*(X2 - X1)
      A2 = Y4 - Y3
      B2 = X3 - X4
      C2 = X3*(Y3 - Y4) + Y3*(X4 - X3)
      A3 = A1
      B3 = Z1 - Z2
      C3 = Z1*(Y1 - Y2) + Y1*(Z2 - Z1)
      A4 = A2
      B4 = Z3 - Z4
      C4 = Z3*(Y3 - Y4) + Y3*(Z4 - Z3)
      X = (B2*C1 - B1*C2) / (A2*B1 - A1*B2)
      Y = (A2*C1 - A1*C2) / (A1*B2 - A2*B1)
      Z = (B4*C3 - B3*C4) / (A4*B3 - A3*B4)
      RETURN
      END
```

```
$IBFTC AV       NODECK
      SUBROUTINE AV (X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, AREA, VOLUME)
      X12 = X1 - X2
      X13 = X1 - X3
      X23 = X2 - X3
      Y12 = Y1 - Y2
      Y13 = Y1 - Y3
      Y23 = Y2 - Y3
      Z12 = Z1 - Z2
      Z13 = Z1 - Z3
      Z23 = Z2 - Z3
      A = SQRT(X12**2 + Y12**2 + Z12**2)
      B = SQRT(X13**2 + Y13**2 + Z13**2)
      C = SCRT(X23**2 + Y23**2 + Z23**2)
      S = (A + B + C) / 2.
      AREA = SQRT(S*(S-A)*(S-B)*(S-C))
      A = Y1 * Z23 - Y2 * Z13 + Y3 * Z12
      B = X1 * Z23 - X2 * Z13 + X3 * Z12
      C = X1 * Y23 - X2 * Y13 + X3 * Y12
      D = X1 * (Y2*Z3 - Y3*Z2) - X2 * (Y1*Z3 - Y3*Z1) +
     + X3 * (Y1*Z2 - Y2*Z1)
      VOLUME = AREA * ABS(D) / SQRT(A**2 + B**2 + C**2) / 3.
      RETURN
      END
```

Identification of Polyhedral Subdivision	*Identification of Subdivision Method*
Frequency 6 Icosahedron	Method 2

No. of vertices for PIT	*No. of edges for PIT*	*No. of faces for PIT*	*No. of vertices for total sphere*	*No. of edges for total sphere*	*No. of faces for total sphere*
V(L) = 28	E(L) = 63	F(L) = 36	V(G) = 362	E(G) = 1080	F(G) = 720

Vertex Identification	*x coordinate*	*y coordinate*	*z coordinate*
0 0	0.000000	0.850651	0.525731
1 0	0.107846	0.758171	0.643075
1 1	0.174499	0.66018	0.468576

= length of element	*Element Identification*	*Unit of length*
Length	1 1 1 0	0.21569298

angle	*angle identification end pt. vertex end pt.*	*degrees*	*radians*
angle Ω Axial	0.0 0.0 1 0 1 1	83.808804	1.462740
angle α Face	1 1 0 0 1 0	71.646295	1.250464
angle α Face	1 1 2 1 1 0	67.513278	1.178329

angle β — Dihedral

element identification dihedral edge — 1 1 1 0

degrees — 175.363857

radians — 3.060677

area, volume, length for one PIT — One Face

	unit area
Area	0.623048
	unit volume
Volume	0.206274
	unit length of elements
Length	12.489358

area, volume, length for total sphere — Complete Structure

	total area
Area	12.460959
	total volume
Volume	4.125473
	total length of elements
Length	216.619799

```
FREQUENCY    6   ICOSAHEDRON     METHOD 2

V(L) =   28   E(L) =   63   F(L) =   36   V(G) =   362   E(G) = 1080   F(G) =   720


  0   0        0.000000        0.850651        0.525731


  1   0        0.107846        0.758171        0.643075

  1   1        0.174499        0.866018        0.468576
     LENGTH           1    1     1    0                       0.21569298
     AXIAL          0.0  0.0     1    0      1    1           83.808804    1.462740
     FACE             1    1     0    0      1    0           71.646295    1.250464
     FACE             1    1     2    1      1    0           67.513278    1.178329
     DIHEDRAL         1    1     1    0                      175.363857    3.060677


  2   0        0.212031        0.639950        0.738585

  2   1        0.291736        0.759807        0.581020
     LENGTH           2    1     2    0                       0.21341332
     AXIAL          0.0  0.0     2    0      2    1           83.874495    1.463886
     FACE             2    1     1    0      2    0           68.618366    1.197616
     FACE             2    1     3    1      2    0           64.858814    1.132000
     DIHEDRAL         2    1     2    0                      174.704821    3.049174

  2   2        0.343074        0.851981        0.395511


  3   0        0.309017        0.500000        0.809017

  3   1        0.401810        0.624919        0.669347
     LENGTH           3    1     3    0                       0.20910066
     AXIAL          0.0  0.0     3    0      3    1           83.998736    1.466055
     FACE             3    1     2    0      3    0           65.240479    1.138661
     FACE             3    1     4    1      3    0           62.105942    1.083953
     DIHEDRAL         3    1     3    0                      174.030888    3.037412

  3   2        0.466634        0.729806        0.499636
```

		LENGTH	3	2	3	1			0.20977452	
		AXIAL	0.0	0.0	3	1	3	2	83.979328	1.465716
		FACE	3	2	2	1	3	1	64.995155	1.134380
		FACE	3	2	4	2	3	1	61.944237	1.081131
		DIHEDRAL	3	2	3	1			173.958036	3.036141
3	3	0.500000			0.809017		0.309017			
4	0	0.395511			0.343074		0.851981			
4	1	0.499636			0.466634		0.729806			
		LENGTH	4	1	4	0			0.20257315	
		AXIAL	0.0	0.0	4	0	4	1	84.186736	1.469336
		FACE	4	1	3	0	4	0	61.609346	1.075286
		FACE	4	1	5	1	4	0	59.236003	1.033863
		DIHEDRAL	4	1	4	0			173.371073	3.025896
4	2	0.577350			0.577350		0.577350			
		LENGTH	4	2	4	1			0.20381459	
		AXIAL	0.0	0.0	4	1	4	2	84.150986	1.468712
		FACE	4	2	3	1	4	1	61.316433	1.070174
		FACE	4	2	5	2	4	1	59.027879	1.030231
		DIHEDRAL	4	2	4	1			173.262043	3.023993
4	3	0.624919			0.669347		0.401810			
4	4	0.639950			0.738585		0.212031			
5	0	0.468576			0.174499		0.866018			
5	1	0.581020			0.291736		0.759807			
		LENGTH	5	1	5	0			0.19408489	
		AXIAL	0.0	0.0	5	0	5	1	84.431107	1.473601
		FACE	5	1	4	0	5	0	57.869432	1.010012
		FACE	5	1	6	1	5	0	56.243358	0.981632
		DIHEDRAL	5	1	5	0			172.752892	3.015107
5	2	0.669347			0.401810		0.624919			
		LENGTH	5	2	5	1			0.19522481	
		AXIAL	0.0	0.0	5	1	5	2	84.398302	1.473028

```
        FACE             5    2        4    1        5    1                 57.502420     1.003607
        FACE             5    2        6    2        5    1                 55.905183     0.975730
        DIHEDRAL         5    2        5    1                              172.611080     3.012632

   5    3          0.729806          0.499636          0.466634
        LENGTH           5    3        5    2                               0.19565051
        AXIAL          0.0  0.0        5    2        5    3                 84.386042     1.472814
        FACE             5    3        4    2        5    2                 57.367121     1.001245
        FACE             5    3        6    3        5    2                 55.788106     0.973686
        DIHEDRAL         5    3        5    2                              172.559191     3.011726

   5    4          0.759807          0.581020          0.291736

   5    5          0.758171          0.643075          0.107846


   6    0          0.525731         -0.000000          0.850651

   6    1          0.643075          0.107846          0.758171
        LENGTH           6    1        6    0                               0.18426310
        AXIAL          0.0  0.0        6    0        6    1                 84.713752     1.478534
        FACE             6    1        5    0        6    0                 54.176847     0.945564
        DIHEDRAL         6    1        6    0                              172.340950     3.007917

   6    2          0.738584          0.212031          0.639950
        LENGTH           6    2        6    1                               0.18426310
        AXIAL          0.0  0.0        6    1        6    2                 84.713747     1.478534
        FACE             6    2        5    1        6    1                 53.512200     0.933964
        DIHEDRAL         6    2        6    1                              172.151842     3.004617

   6    3          0.809017          0.309017          0.500000
        LENGTH           6    3        6    2                               0.18426311
        AXIAL          0.0  0.0        6    2        6    3                 84.713752     1.478534
        FACE             6    3        5    2        6    2                 53.150170     0.927645
        DIHEDRAL         6    3        6    2                              172.047482     3.002795

   6    4          0.851901          0.395511          0.343074

   6    5          0.866018          0.468576          0.174499

   6    6          0.850651          0.525731         -0.000000
```

ONE FACE	
AREA	0.623048
VOLUME	0.206274
LENGTH	12.489358
COMPLETE STRUCTURE	
AREA	12.460959
VOLUME	4.125473
LENGTH	216.619799

APPENDIX II

The plot routine was written for the IBM 7044 computer utilizing FORTRAN language. The program generates data necessary for input tapes to drive a CalComp plotter model 470.

The example of the output map is for a six frequency Icosahedral sphere oriented in three directions as shown in Figures 15, 16, and 17.

```
$IBFTC PTTOI    NODECK
      DIMENSION COORD(3, 3, 32), A(3, 3, 8), B(3, 3, 8), F(3, 3, 8),
     1  D(3, 3, 8), BUFF(4000), ALPHA(8), V(3, 3), LAB(2, 3), P(3, 3),
     2  SD 1(3, 32), SD 2(3, 32), SD 3(3, 32), EX(768, 3), EXP(768, 3),
     3  C(3, 3)
      COMMON DC(6, 3), X2R2, Y2R2, Z2R2, X2R1, Y2R1, X1R1
      EQUIVALENCE (COORD(1, 1, 1), A), (COORD(1, 1, 9), B),
     1  (COORD(1, 1, 17), F), (COORD(1, 1, 25), D),
     2  (V(1, 1), X1), (V(2, 1), Y1), (V(3, 1), Z1),
     3  (V(1, 2), X2), (V(2, 2), Y2), (V(3, 2), Z2),
     4  (V(1, 3), X3), (V(2, 3), Y3), (V(3, 3), Z3),
     5  (LAB(1, 1), I1), (LAB(2, 1), J1), (LAB(1, 2), I2),
     6  (LAB(2, 2), J2), (LAB(1, 3), I3), (LAB(2, 3), J3)
      LOGICAL OUT
      INTEGER RU, TU, RAD, TE OC IC, Q
      COEF(A1, A2, A3, B1, B2, B3) = A1 * (B2 - B3) - A2 * (B1 - B3) +
     1  A3 * (B1 - B2)
      DATA A   /    -.57735027, -.57735027,  .57735027,  .57735027,
     1  -.57735027, -.57735027, -.57735027,  .57735027, -.57735027,
     2  -.57735027, -.57735027,  .57735027,  .57735027,  .57735027,
     3   .57735027,  .57735027, -.57735027, -.57735027, -.57735027,
     4  -.57735027,  .57735027, -.57735027,  .57735027, -.57735027,
     5   .57735027,  .57735027,  .57735027,  .57735027,  .57735027,
     6   .57735027, -.57735027,  .57735027, -.57735027,  .57735027,
     7  -.57735027, -.57735027,1.,0.,0.,0.,0.,1.,0.,1.,0.,1.,0.,0.,0.,
     8  1.,0.,0.,0.,-1.,1.,0.,0.,0.,0.,-1.,0.,-1.,0.,
     9  1.,0.,0.,0.,-1.,0.,0.,0.,1./
      DATA B / -1., 0.,0.,0.,0.,1.,0.,-1.,0.,
     1  -1.,0.,0.,0.,1.,0.,0.,0.,1.,
     2  -1.,0.,0.,0.,0.,-1.,0.,1.,0.,
     3  -1.,0.,0.,0.,-1.,0.,0.,0.,-1.,
     40.,.85065081,.52573111,.52573111,0.,.85065081,.85065081,
     5.52573111,0.,0.,-.85065081,.52573111,.85065081,-.52573111,0.,
     6.52573111,0.,.85065081,0.,-.85065081,.52573111,-.52573111,0.,
     7.85065081,-.85065081,-.52573111,0.,0.,.85065081,.52573111,
     8-.85065081,.52573111,0.,-.52573111,0.,.85065081 /
      DATA F / 0., .85065081,-.52573111, .85065081, .52573111, 0., .5257
     13111,0.,-.85065081,0.,-.85065081,-.52573111,.52573111,0.,-.8506508
     21,.85065081,-.52573111,0.,0.,-.85065081,-.52573111,-.85065081,-.52
     3573111,0.,-.52573111,0.,-.85065081,0.,.85065081,-.52573111,-.52573
     4111,0.,-.85065081,-.85065081,.52573111,0.,0.,.85065081,.52573111,-
     5.52573111,0.,.85065081,.52573111,0.,.85065081,0.,-.85065081,.52573
     6111,.52573111,0.,.85065081,-.52573111,0.,.85065081,0.,.85065081,-.
     752573111,.52573111,0.,-.85065081,-.52573111,0.,-.85065081,0.,-.850
     865081,-.52573111,-.52573111,0.,-.85065081,.52573111,0.,-.85065081/
      DATA D / 0., .85065081, .52573111, .85065081, .52573111,0.,0.,.850
     165081,-.52573111,0.,.85065081,.52573111,0.,.85065081,-.52573111,-.
     285065081,.52573111,0.,0.,-.85065081,.52573111,0.,-.85065081,-.5257
     33111,.85065081,-.52573111,0.,0.,-.85065081,.52573111,-.85065081,-.
     452573111,0.,0.,-.85065081,-.52573111,.85065081,.52573111,0.,.52573
     5111,0.,.85065081,.85065081,-.52573111,0.,.85065081,.52573111,0.,.8
     65065081,-.52573111,0.,.52573111,0.,-.85065081,-.85065081,.52573111
     7,0.,-.85065081,-.52573111,0.,-.52573111,0.,.85065081,-.85065081,.5
     82573111,0.,-.52573111,0.,-.85065081,-.85065081,-.52573111,0. /
      DATA RAD / 1HR /
```

```
      CALL PLOTS (BUFF, 4000)
      CALL PLOT (0., 11.5, 23)
      CALL PLOT (0., -11., 21)
      CALL PLOT (6., -.5, 23)
    1 READ (5, 70) N, TE OC IC, ROT, RU, TILT, TU, ALPHA, METHOD
   70 FORMAT (2I2, 6X, 2(F9.0, A1), 8A6, I2)
      IF (N .EQ. 0) STOP
      CALL PLOT (0., 5.5, 23)
      CALL SYMBOL (-5., -4.50, .14, 10HFREQUENCY , 0., 10)
      CALL NUMBER (-0., -0., .14, FLOAT(N), 0., -1)
      CALL SYMBOL (-5., -4.75, .14, 9HROTATION , 0., 9)
      CALL NUMBER (-0., -0., .14, ROT, 0., 3)
      IF (RU .NE. RAD) GO TO 112
      CALL SYMBOL (-0., -0., .14, 8H RADIANS, 0., 8)
      GO TO 113
  112 CALL SYMBOL (-0., -0., .14, 8H DEGREES, 0., 8)
  113 CALL SYMBOL (-5., -5.00, .14, 5HTILT , 0., 5)
      CALL NUMBER (-0., -0., .14, TILT, 0., 3)
      IF (TU .NE. RAD) GO TO 114
      CALL SYMBOL (-0., -0., .14, 8H RADIANS, 0., 8)
      GO TO 115
  114 CALL SYMBOL (-0., -0., .14, 8H DEGREES, 0., 8)
  115 CALL SYMBOL (-5., -5.25, .14, ALPHA, 0., 48)
      IF (RU .NE. RAD) ROT = ROT * .017453293
      IF (TU .NE. RAD) TILT = TILT * .017453293
      CALL DATE (4., -5., .14)
      CALL FACTOR (5.)
      CALL SYMBOL (-1.2, -1., .05, 3, 0., -1)
      FLOAT N = N
      SIN R = SIN(ROT)
      COS R = COS(ROT)
      SIN T = SIN(TILT)
      COS T = COS(TILT)
      MIN = 1
      MAX = 4
      IF (TE OC IC .NE. 8) GO TO 2
      MIN = 5
      MAX = 12
    2 IF (TE OC IC .NE. 20) GO TO 3
      MIN = 13
      MAX = 32
    3 DO 4 I = 1, 3
      DO 4 J = 1, 3
    4 V(J, I) = COORD(J, I, MIN)
      L1 = 0
      DO 7 L2 = 1, 2
      DO 7 LN = 2, 3
      IF (L2 .EQ. LN) GO TO 7
      L1 = L1 + 1
      CALL RCTATE (1, V(1, LN), V(2, LN), V(3, LN), V(1, L2), V(2, L2),
     * V(3, L2))
      THETA = ATAN2(Y2R1, X2R1)
      DO 7 L4 = 1, N
      T = L4
      ANG = T * THETA / FLOAT N
```

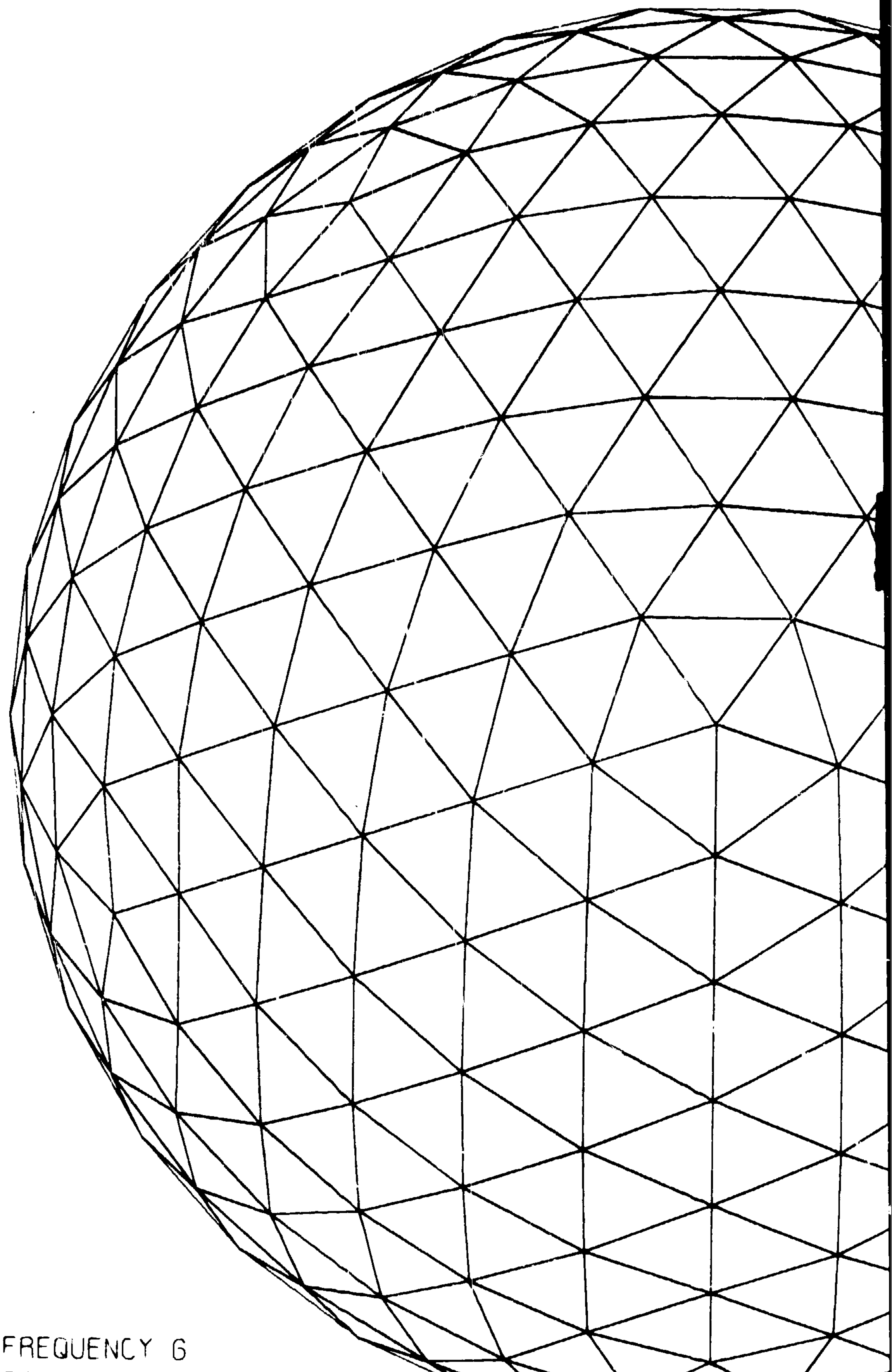
```
      X = COS(ANG)
      Y = SIN(ANG)
      A1 = -Y
      B1 = X
      A2 = -Y2R1
      B2 = X2R1 - 1.
      C2 = B2
      X = (B1*C2) / (A2*B1 - A1*B2)
      Y = (A1*C2) / (A1*B2 - A2*B1)
      CALL RCTATE (3, X, Y, 0.)
      IF(L1 - 2) 45, 5, 6
   45 SD 1(1, L4) = X2R2
      SD 1(2, L4) = Y2R2
      SD 1(3, L4) = Z2R2
      GO TO 7
    5 SD 2(1, L4) = X2R2
      SD 2(2, L4) = Y2R2
      SD 2(3, L4) = Z2R2
      GO TO 7
    6 SD 3(1, L4) = X2R2
      SD 3(2, L4) = Y2R2
      SD 3(3, L4) = Z2R2
    7 CONTINUE
      Q = 0
      DO 16 I1 = 1, N
      DO 16 J1 = 1, I1
      I2 = I1 - 1
      J2 = J1 - 1
      I3 = I1
      J3 = J2
      OUT = .FALSE.
    8 DO 13 LA = 1, 3
      LI = LAB(1, LA)
      LJ = LAB(2, LA)
      IJ = LI - LJ
      NJ = N - LJ
      IF (LI .EQ. LJ) GO TO 9
      IF (LJ .EQ. 0) GO TO 11
      IF (LI .NE. N) GO TO 12
      P(LA, 1) = SD 3(1, NJ)
      P(LA, 2) = SD 3(2, NJ)
      P(LA, 3) = SD 3(3, NJ)
      GO TO 13
    9 IF (LJ .EQ. 0) GO TO 10
      P(LA, 1) = SD 1(1, LI)
      P(LA, 2) = SD 1(2, LI)
      P(LA, 3) = SD 1(3, LI)
      GO TO 13
   10 P(LA, 1) = X1
      P(LA, 2) = Y1
      P(LA, 3) = Z1
      GO TO 13
   11 P(LA, 1) = SD 2(1, LI)
      P(LA, 2) = SD 2(2, LI)
      P(LA, 3) = SD 2(3, LI)
```

```
      GO TO 13
   12 CALL INTERC (SD 1(1, LI), SD 1(2, LI), SD 1(3, LI),SD2(1, LI),
     1  SD 2(2, LI), SD 2(3, LI), SD 3(1, NJ), SD 3(2, NJ), SD 3(3, NJ),
     2  SD 1(1, LJ), SD 1(2, LJ), SD 1(3, LJ), C(1, 1), C(1, 2), C(1,3))
      CALL INTERC (SD 3(1, IJ), SD 3(2, IJ), SD 3(3, IJ), SD 2(1, IJ),
     1  SD 2(2, IJ), SD 2(3, IJ), SD 3(1, NJ), SD 3(2, NJ), SD 3(3, NJ),
     2  SD 1(1, LJ), SD 1(2, LJ), SD 1(3, LJ), C(2, 1), C(2, 2), C(2,3))
      CALL INTERC (SD 3(1, IJ), SD 3(2, IJ), SD 3(3, IJ), SD 2(1, IJ),
     1  SD 2(2, IJ), SD 2(3, IJ), SD 1(1, LI), SD 1(2, LI), SD 1(3, LI),
     2  SD 2(1, LI), SD 2(2, LI), SD 2(3, LI), C(3, 1),C(3, 2), C(3,3))
      GO TO (1, 99, 100), METHOD
   99 P(LA, 1) = C(1, 1) + C(2, 1) + C(3, 1)
      P(LA, 2) = C(1, 2) + C(2, 2) + C(3, 2)
      P(LA, 3) = C(1, 3) + C(2, 3) + C(3, 3)
      GO TO 13
  100 DO 101 L = 1, 3
      DENOM=SQRT(C(L,1)**2+C(L,2)**2+C(L,3)**2)
      DO 101 M=1,3
  101 C(L,M)=C(L,M)/DENOM
C     VERTICES OF WINDOW HAVE BEEN EXPLODED TO SURFACE OF SPHERE.
      DO 102 L=1,2
      DO 102 M=1,3
  102 C(L,M)=C(L,M)-C(3,M)
C     WINDOW HAS BEEN TRANSLATED.8VERTEX C,I.E. VERTEX 3 IS AT 0,0,0.
      CALL ROTATE (1, C(1, 1), C(1, 2), C(1, 3), C(2, 1), C(2, 2),
     *  C(2, 3))
      ANG = .5 * ATAN2(Y2R1,X2R1)
      XR1 = X2R1* COS(ANG) + Y2R1* SIN(ANG)
      YR1 = Y2R1* COS(ANG) - X2R1* SIN(ANG)
      ANG = ATAN2(Y2R1,X1R1- X2R1)
      ANG = .5 * ANG
      XR2 = (X2R1- X1R1)* COS(ANG) - Y2R1* SIN(ANG) + X1R1
      YR2 = Y2R1* COS(ANG) + (X2R1- X1R1)* SIN(ANG)
      S1 = YR1 / XR1
      S2 = YR2 / (XR2 - X1R1)
      C2 = X1R1* S2
      D9= - S1 + S2
      CX = C2 / D9
      CY = (S1 * C2) / D9
      CALL ROTATE (3, CX, CY, 0.0)
C     CENTER OF WINDOW HAS BEEN FOUND. TRANSLATE BACK. . .8
      P(LA, 1) = X2R2+ C(3, 1)
      P(LA, 2) = Y2R2+ C(3, 2)
      P(LA, 3) = Z2R2+ C(3, 3)
   13 CONTINUE
      DO 14 IN1 = 1, 3
      R = SQRT(P(IN1, 1)**2 + P(IN1, 2)**2 + P(IN1, 3)**2)
      Q = Q + 1
      DO 14 IN2 = 1, 3
   14 EX (Q, IN2) = P(IN1, IN2) / R
      I3 = I2
      J2 = J2 + 1
      IF (OUT) GO TO 16
      IF (J2 .GT. I2) GO TO 16
      OUT = .TRUE.
```

```
      GO TO 8
   16 CONTINUE
      WRITE (6, 73) Q
   73 FORMAT (5H Q = I4)
      TX1 = X1 - X3
      TY1 = Y1 - Y3
      TZ1 = Z1 - Z3
      TX2 = X2 - X3
      TY2 = Y2 - Y3
      TZ2 = Z2 - Z3
      DO 17 K = 1, Q
      DO 17 K1 = 1, 3
   17 EXP(K, K1) = EX(K, K1) - V(K1, 3)
      CALL ROTATE (1, TX2, TY2, TZ2, TX1, TY1, TZ1)
      DO 23 M = MIN, MAX
      IF (M .EQ. MIN) GO TO 21
      DO 18 I = 1, 3
      DO 18 J = 1, 3
   18 V(J, I) = COORD(J, I, M)
      TX1 = X1 - X3
      TY1 = Y1 - Y3
      TZ1 = Z1 - Z3
      TX2 = X2 - X3
      TY2 = Y2 - Y3
      TZ2 = Z2 - Z3
      CALL ROTATE (2, TX2, TY2, TZ2, TX1, TY1, TZ1)
      DO 19 K = 1, 3
      DO 19 I = 1, 3
   19 P(I, K) = DC(1, K)*DC(4, I) + DC(2, K)*DC(5, I) + DC(3, K)*DC(6,I)
      DO 20 K = 1, Q
      DO 20 L = 1, 3
   20 EX(K,L) =EXP(K,1)*P(L,1)+EXP(K,2)*P(L,2)+EXP(K,3)*P(L,3) + V(L,3)
   21 DO 22 K1 = 1, Q
      YRT = EX(K1,2) * COS R - EX(K1,1) * SIN R
      ZRT = EX(K1,3) * COS T - EX(K1,1) * COS R * SIN T - EX(K1, 2) *
     *  SIN R * SIN T
      EX(K1, 2) = YRT
   22 EX(K1, 3) = ZRT
      DO 23 K1 = 1, Q, 3
      K2 = K1 + 1
      K3 = K1 + 2
      IF (COEF(EX(K1, 2), EX(K2, 2), EX(K3, 2), EX(K1, 3), EX(K2, 3),
     *  EX(K3, 3)) .LE. 0.) GO TO 23
      CALL PLOT (EX(K1, 2), EX(K1, 3), 3)
      CALL PLOT (EX(K2, 2), EX(K2, 3), 2)
      CALL PLOT (EX(K3, 2), EX(K3, 3), 2)
      CALL PLOT (EX(K1, 2), EX(K1, 3), 2)
      WRITE (6, 705) (EX(NY, 2), EX(NY, 3), NY = K1, K3)
  705 FORMAT (1X, 6F14.6)
   23 CONTINUE
      CALL SYMBOL (-1.2, -1., .03, 3, 0., -1)
      CALL FACTOR (1.)
      CALL PLOT (8.875, 11., 3)
      CALL PLOT (8.875, 0., 1)
      CALL PLOT (14.875, -.5, -3)
```

```
$IBFTC ROTATE  NODECK
      SUBROUTINE ROTATE (N, X2, Y2, Z2, X1, Y1, Z1)
      COMMON DC(6, 3), X2R2, Y2R2, Z2R2, X2R1, Y2R1, X1R1
      IF (N .EQ. 3) GO TO 1
      M1 = 3*N - 2
      M2 = M1 + 1
      M3 = M2 + 1
      D1 = SQRT(X1**2 + Y1**2 + Z1**2)
      DC(M1, 1) = X1 / D1
      DC(M1, 2) = Y1 / D1
      DC(M1, 3) = Z1 / D1
      H3 = Y1 * Z2 - Y2 * Z1
      U3 = Z1 * X2 - X1 * Z2
      V3 = X1 * Y2 - X2 * Y1
      D2 = SQRT(H3**2 + U3**2 + V3**2)
      DC(M3, 1) = H3 / D2
      DC(M3, 2) = U3 / D2
      DC(M3, 3) = V3 / D2
      H2 = DC(M3, 2) * Z1 - Y1 * DC(M3, 3)
      U2 = DC(M3, 3) * X1 - Z1 * DC(M3, 1)
      V2 = DC(M3, 1) * Y1 - X1 * DC(M3, 2)
      D3 = SQRT(H2**2 + U2**2 + V2**2)
      DC(M2, 1) = H2 / D3
      DC(M2, 2) = U2 / D3
      DC(M2, 3) = V2 / D3
      X1R1 = X1*DC(M1, 1) + Y1*DC(M1, 2) + Z1*DC(M1, 3)
      X2R1 = X2*DC(M1, 1) + Y2*DC(M1, 2) + Z2*DC(M1, 3)
      Y2R1 = X2*DC(M2, 1) + Y2*DC(M2, 2) + Z2*DC(M2, 3)
      RETURN
    1 X2R2 = X2*DC(M1, 1) + Y2*DC(M2, 1) + Z2*DC(M3, 1)
      Y2R2 = X2*DC(M1, 2) + Y2*DC(M2, 2) + Z2*DC(M3, 2)
      Z2R2 = X2*DC(M1, 3) + Y2*DC(M2, 3) + Z2*DC(M3, 3)
      RETURN
      END
```

```
$IBFTC INTERC  NODECK
      SUBROUTINE INTERC (X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3, X4, Y4, Z4,
     *  X, Y, Z)
      A1 = Y2 - Y1
      B1 = X1 - X2
      C1 = X1*(Y1 - Y2) + Y1*(X2 - X1)
      A2 = Y4 - Y3
      B2 = X3 - X4
      C2 = X3*(Y3 - Y4) + Y3*(X4 - X3)
      A3 = A1
      B3 = Z1 - Z2
      C3 = Z1*(Y1 - Y2) + Y1*(Z2 - Z1)
      A4 = A2
      B4 = Z3 - Z4
      C4 = Z3*(Y3 - Y4) + Y3*(Z4 - Z3)
      X = (B2*C1 - B1*C2) / (A2*B1 - A1*B2)
      Y = (A2*C1 - A1*C2) / (A1*B2 - A2*B1)
      Z = (B4*C3 - B3*C4) / (A4*B3 - A3*B4)
      RETURN
      END
```



FREQUENCY 6
ROTATION 0.000 DEGREES
TILT 58.282 DEGREES
ORIENTATION VERTICE

FOLDOUT FRAME 1

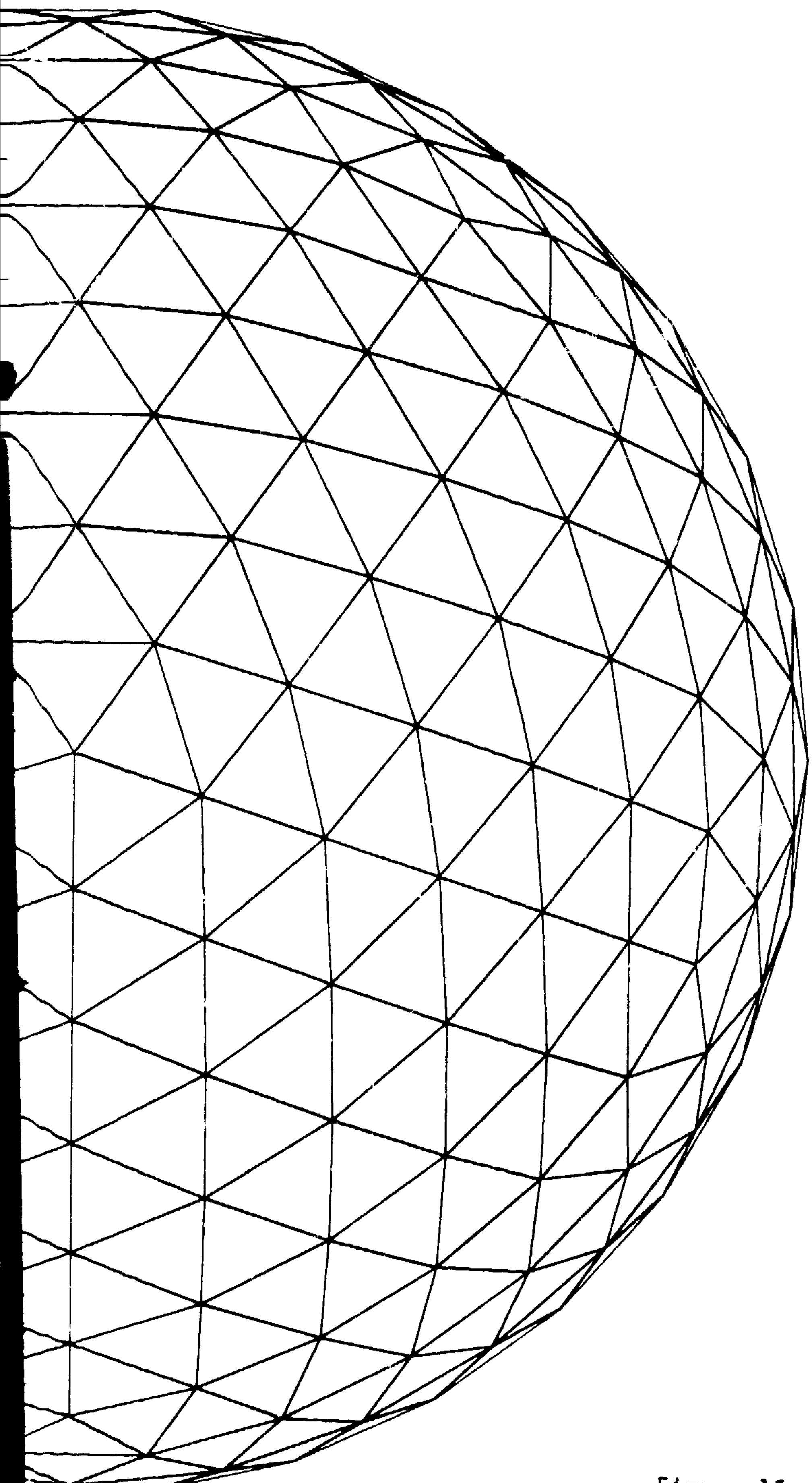
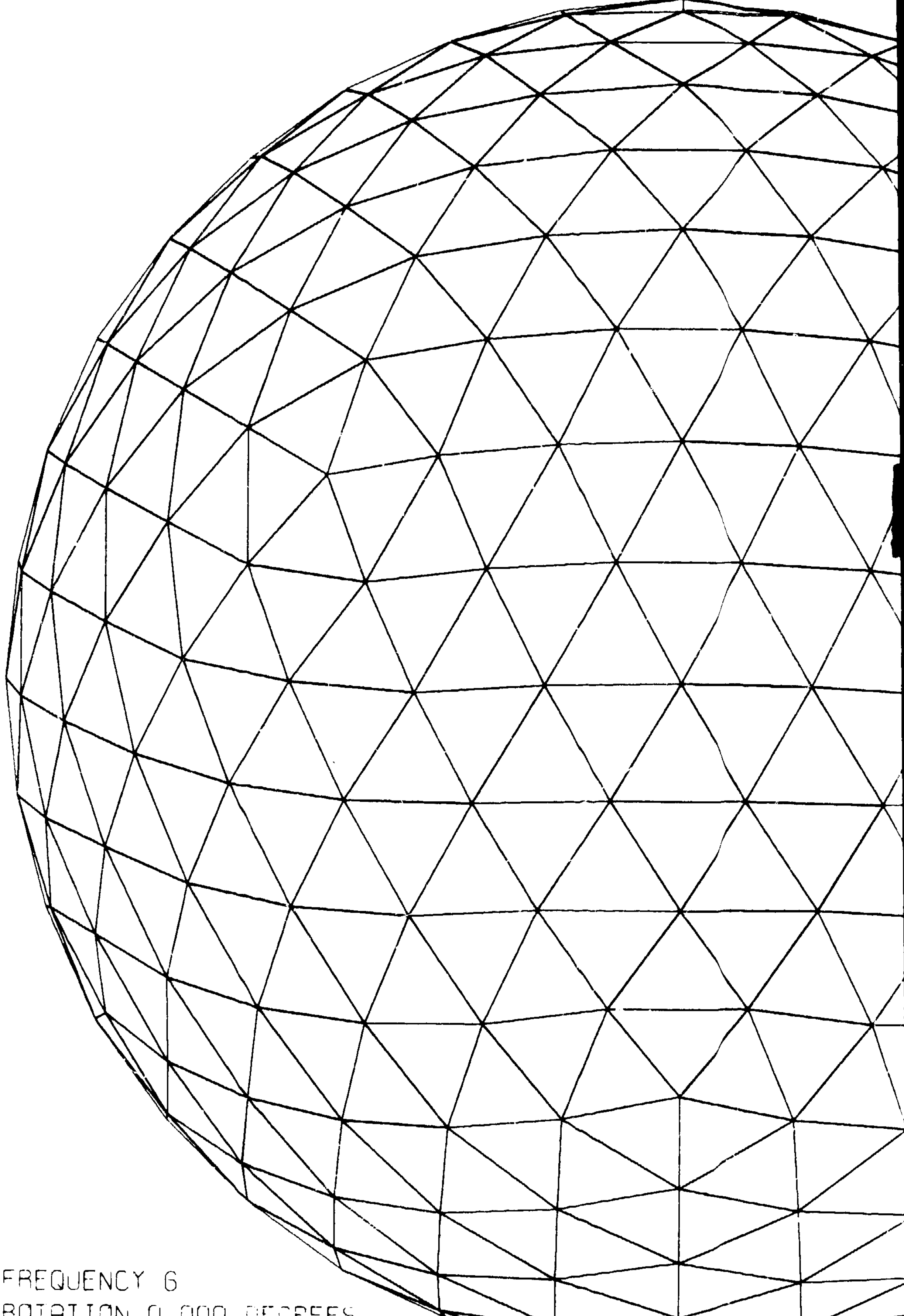
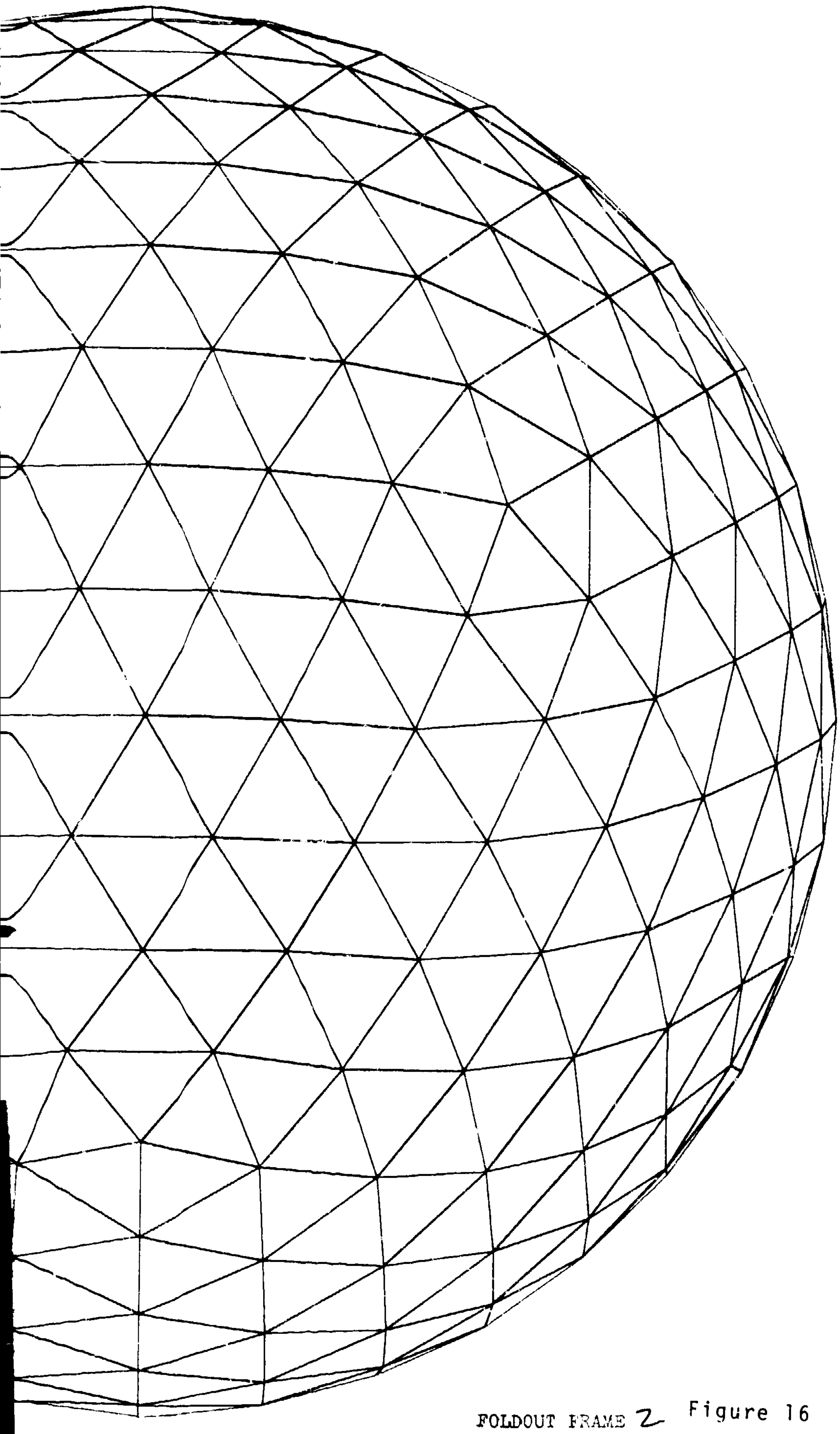


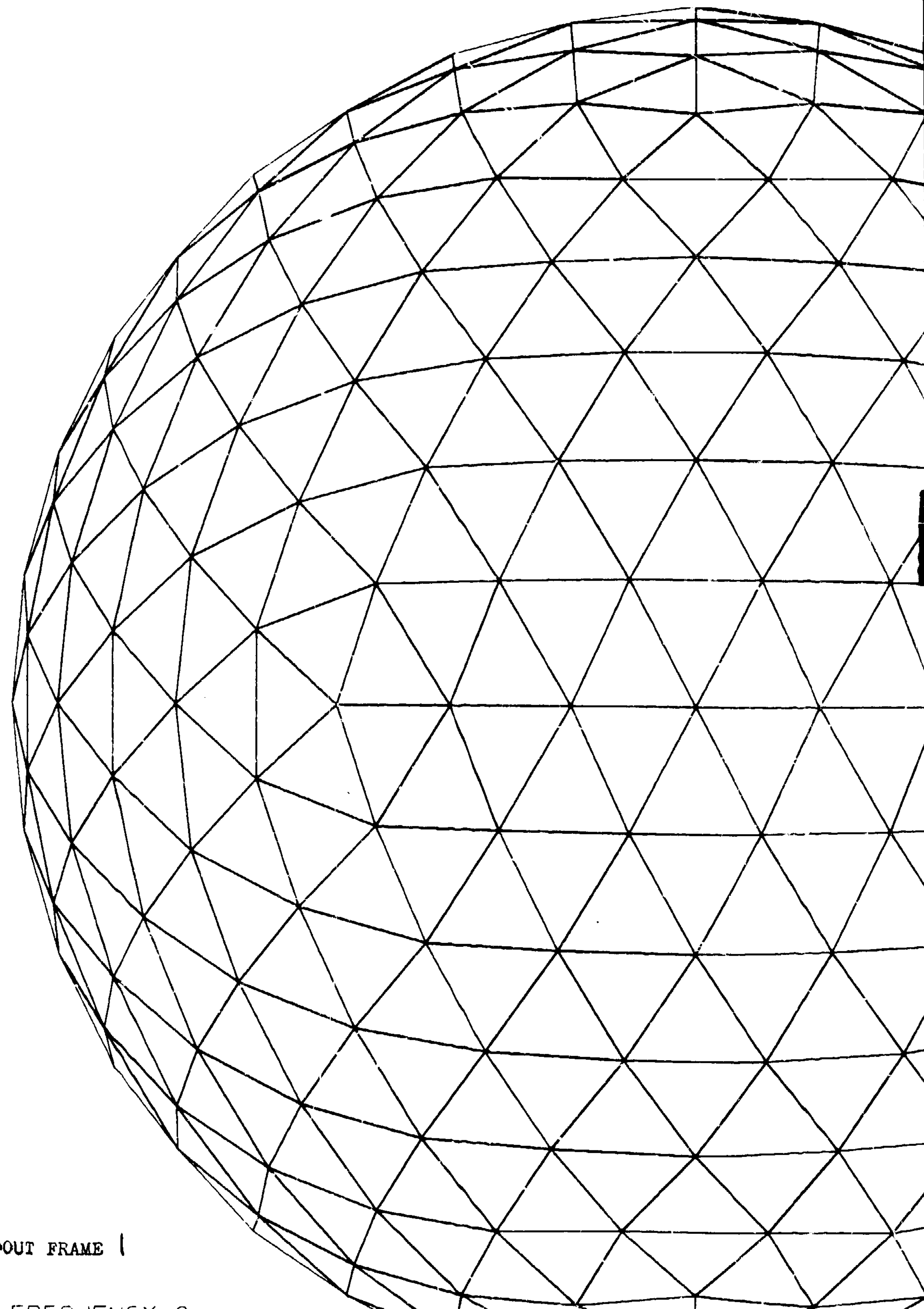
Figure 15 FOLDOUT FRAME 2



FREQUENCY 6
ROTATION 0.000 DEGREES
FOLDOUT FRAME | TILT 20.905 DEGREES
ORIENTATION FACE

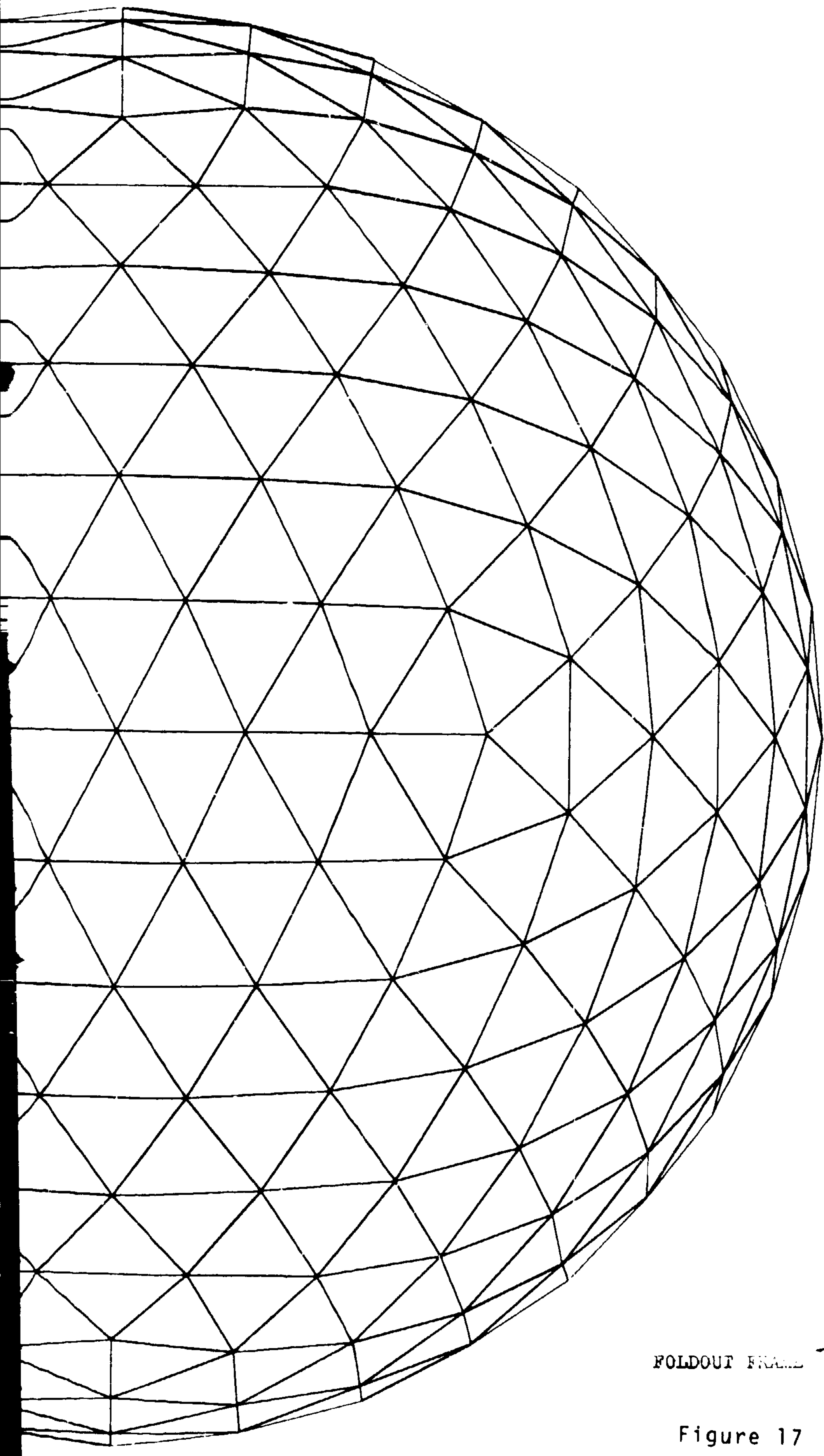


FOLDOUT FRAME 2 Figure 16



FOLDOUT FRAME 1

FREQUENCY 6
ROTATION 0.000 DEGREES
TILT 0.000 DEGREES
ORIENTATION EDGE



FOLDOUT FRAME 2

Figure 17