UTEC-CSc-78-135

DISPLAY OF COMPLEX THREE DIMENSIONAL FINITE ELEMENT MODELS

Contractor:	University of Utah
Effective Date:	1 May 1975
Expiration Date:	30 September 1977
Reporting Period:	1 May 1975 - 30 September 1977

Principal Investigator:
Dr. David C. Evans
Professor of Computer Science
College of Engineering
University of Utah
Salt Lake City, Utah

Consultants:
Dr. Henry N. Christiansen
Professor of Civil Engineering
College of Engineering Science & Technology
Brigham Young University
Provo, Utah

Thomas W. Sederberg
College of Engineering Science & Technology
Brigham Young University
Provo, Utah

Final Report
for

OFFICE OF NAVAL RESEARCH
CODE 439
NAVY DEPARTMENT
ARLINGTON, VIRGINIA 22217
Under Contract No. N00014-75-C-0194

April 1978

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
UTEC-CSc-78-135

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (*and Subtitle*)
"Display of Complex Three Dimensional Finite Element Models"

5. TYPE OF REPORT & PERIOD COVERED
Final Technical Report
1 May 1975 - 30 Sept. 1977

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(*s*)
David Evans, Henry Christiansen,
Thomas W. Sederberg

8. CONTRACT OR GRANT NUMBER(*s*)
N000145-75-C-0194

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Utah
Computer Science Department
Salt Lake City, Utah 84112

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Task No: RN-64-1554

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of Naval Research
Arlington, Virginia 22217

12. REPORT DATE
April 1978

13. NUMBER OF PAGES
94

14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)
Office of Naval Research
University of California
553 Evans Hall
Berkeley, CA 94720

15. SECURITY CLASS. (*of this report*)
unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)
Finite element, Contour line definition, Panel definition, Digitized contour line, Node elimination, Triangulation, Triangulation algorithm, Mapping procedure, Mutually centered, Quadrilateral formation, User interaction, Network of discrete polygonal elements, Connectivity of nodes, Triangulation ambiguity, Topographical branching

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

Complex three dimensional models can be displayed after an automatic generation of a finite element (panel) mapping. Although this automatic generation algorithm fails at certain levels of model complexity, the elimination of these failures can be accomplished through user interaction. This report presents the algorithm solution to the problem of converting a contour definition of an arbitrary surface into a panel definition. The algorithm has been rigorously tested and experience with a highly complex data base lends credence to the claim of a general solution. Future work

FORWARD

This document represents a final report to the Office of Naval Research on Task No. NR 064-554, the Display of Complex Three Dimensional Finite Element Models - Phase III. The main body of the report is a discussion of the computer programs which have been developed during Phase III. This material was prepared by Thomas W. Sederberg and serves as his M. S. Thesis at Brigham Young University. The remaining paragraphs in this forward discuss activities related to the contract that are not discussed in the main body of the report.

Distribution of MOVIE.BYU

This general purpose computer graphics software package (largely the result of efforts under Phases I and II of this contract), has now been distributed to approximately 160 organizations in the United States, Canada, England, France, Germany, Norway, Israel, and Australia. A complete mailing list of those organizations is included as Appendix C.

Technical Presentations

Since the award of Phase III of this contract, technical presentations featuring results obtained under this funding have been made by Dr. Christiansen to the following groups. Of course, travel funds came from many sources and some of the presentations were on an informal basis.

1. Envirotech Corporation (Winston-Salem, North Carolina)
2. Waterways Experiment Station, Corps of Engineers (Vicksburg, Miss.)
3. University and College Designers Association - National Conference (Provo, Utah)

4. Engineering Society of Detroit's Third Annual Computer Graphics Conference (Detroit, Michigan)
5. Industrial Design Department, Center for Creative Studies (Detroit, Michigan)
6. Computer Graphics Workshop, University of Arizona (Tucson, Arizona)
7. Graduate Seminar in Computer Science, Brigham Young University (Provo, Utah)
8. 3rd Southwest Graduate Research Conference in Applied Mechanics, University of Texas (Austin, Texas)
9. Department of Creative Arts, Purdue University (Lafayette, Indiana)
10. Raytheon Missile Systems (Bedford, Mass.)
11. Graphics Utah Style - 77 (Snowbird, Utah)
12. Symposium on Computer Methods in Engineering, University of Southern California (Los Angeles, California)
13. Genisco Corporation (Irvine, California)
14. Tektronix Corporation (Wilsonville, Oregon)
15. Shell Development Co. (Houston, Texas)
16. Design and Drafting Seminar, Brigham Young University (Provo, Utah)
17. American Society of Mechanical Engineers, Annual Meeting (Atlanta, Georgia)
18. Art Directors Club (Salt Lake City, Utah)

Since the final report for Phase II, the following technical papers have been published.

Christiansen, H. N., Brown, B. E., and McCleary, L. E., "A General Purpose Computer Graphics Display System for Finite Element Models," 46th Shock and Vibration Bulletin, Part 5, August 1976, pp. 61-66.

Christiansen, H. N., "Computer Graphics - Treatment for the Terminal Illness," Preprint 2765 - Development of Computational Methods in Structural Analysis and Design: Past, Present, and Future, ASCE, Philadelphia, PA, Sept. 1976, pp. VI 1-11.

Christiansen, H. N., "Computer Simulation of Distorted Structural Frameworks," Journal of Computers and Structures, Vol. 6, Dec. 1976, pp. 497-501.

Christiansen, H. N., and Stephenson, M. B., "MOVIE.BYU - A General Purpose Computer Graphics Display System," Proceedings of the Symposium on Applications of Computer Methods in Engineering, USC, Aug. 1977.

Recently, three more technical papers (two covering aspects of Phases I and II and other concerned with Phase III) have been written and submitted for upcoming meetings.

TABLE OF CONTENTS

FORWARD ii

LIST OF TABLES vi

LIST OF FIGURES vii

Chapter

1. INTRODUCTION 1

2. AN OVERVIEW OF COMPUTER GRAPHICS 3

3. CONVERTING CONTOURS INTO PANELS 6

    A LIMITED TRIANGULATION ALGORITHM 6

    MAPPING 9

    ULTIMATE AMBIGUITY 12

    BRANCHING 15

4. BRAIN CONTOUR DATA 20

    ORIGIN AND DESCRIPTION 20

    BRAIN DATA FORMAT TRANSMUTATION 23

5. ECONOMIZING 26

    NODE ELIMINATION 26

    QUADRILATERAL FORMATION 27

6. FORTRAN IMPLIMENTATION 29

7. EXAMPLE PROBLEM 31

8. PICTURES 41

9. CONCLUSIONS 48

BIBLIOGRAPHY 49

APPENDIX A. COMPUTER PROGRAM 50

APPENDIX B. USER DOCUMENTATION 73

APPENDIS C. MOVIE.BYU MAILING LIST 79

LIST OF TABLES

Table Page

1. Index to Brain Data 22

LIST OF FIGURES

Figure Page

1. Contour Pair Prior to Triangulation 7
2. Triangulated Contour Pair 8
3. Commencing Triangulation 9
4. Failure Example 10
5. Window Parameters 11
6. Mapped Contour Pair 12
7. Synonimous Triangulation Interpretations 13
8. Non-Synonimous Triangulation Interpretations 14
9. Simple Case of Branching 15
10. Uneconomical Handling of Branching 16
11. Preceding View with Hidden Surface Elimination 16
12. Preferred Handling of Branching 17
13. Typical Problem in Connectivity 18
14. Overlap Test 19
15. Contour Line of Brain Cortex 21
16. Segment Definition 23
17. Determining Direction of Rotation 24
18. Brain Contour Error 25
19. Node Elimination Parameters 26
20. Node Reduction Flow Chart 26
21. Warp Angle Determination 28
22. Caudates 31
23. Computer Dialogue 32

Figure Page

24. Continuous Tone Rendering of Caudates 34

25. Example 36

26. Example 37

27. Example 38

28. Example 39

29. Example 40

30. Brain Stem - 1991 Panels 43

31. Brain Stem - 492 Panels 43

32. Labelled Composite View 44

33. Composite View Line Drawing 45

34. Brain Stem with 1991 Panels 46

35. Brain Stem with 492 Panels 46

36. Thalamus with Flat Shading 46

37. Thalamus with Smooth Shading 46

38. Composite View 47

39. Cortex Slice 47

40. Mount Timpanogos - Flat Shading 47

41. Mount Timpanogos - Smooth Shading 47

Chapter 1

INTRODUCTION

There has been a disparity between the conventional method of describing topographic surfaces (i.e. contour line definition) and a format of surface description often used in continuous-line computer graphics (i.e. panel definition). The two differ enough that conversion from contours to panels is not a trivial problem. A computer program that performs such a conversion would greatly facilitate continous tone display of topographical surfaces, or any other surface which is defined by contour lines.

This problem has been addressed by Keppel[1] and alluded to by Fuchs[2]. Keppel's is an highly systematic approach in which he uses graph theory to find the panel arrangement which maximizes the volume enclosed by concave surfaces. Fuchs mentions an approach to the problem as part of an algorithm to reconstruct a surface from data retrieved from a laser scan sensor.

This thesis elaborates on a general conversion system. Following a brief overview of computer graphics, a simple algoritm

---

[1] E. Keppel, "Approximating Complex Surfaces by Triangulation of Contour Lines," Journal of Research and Development, IBM Vol. 19, No. 1 (January 1975), 2-11.

[2] Henry Fuchs, "The Automatic Sensing of 3-Dimensional Surface Points from Visual Scenes" (unpublished PhD dissertation, University of Utah, 1975.)

is described which extracts a panel definition from a pair of adjacent contour loops subject to the restriction that the two loops are similarly sized and shaped, and are mutually centered. Next, a mapping procedure is described which greatly relaxes the above restrictions. It is also shown that the conversion from contours to panels is inherently ambiguous (to various degrees) and that occasionally the amibiguity is great enough to require user interaction to guide the conversion algorithm. An important complication addressed in this thesis is the problem of handling cases where one contour loop branches into two or more (or vice versa).

Attention turns next to a contour line definition of the human brain, and special problems encountered in preparing those data for continuous tone display. The final chapters explain the fortran implementation, present an example problem, and show sample pictures of the brain parts.

Chapter 2

AN OVERVIEW OF COMPUTER GRAPHICS

The past decade has seen fantastic advances in the field of computer graphics. Today, it is a sheltered person who is not familiar with some form of computer graphics, be it Snoopy calendars or computer ping-pong on one end of the spectrum, or sophisticated airline pilot training simulators on the other end. Display mediums used in graphics are very diverse, and include raster scan cathode ray tubes, cathode ray storage tubes, conventional line printers, plotting machines, and film recorders. Perhaps the most life-like pictures are continuous tone images produced on raster scan cathode ray tubes.

Continuous tone display requires the capability of defining the light intensity of each pixel of a scan line - TV style. There are typically 512 scan lines per picture with 512 pixels per line, and 256 levels of light intensity for each pixel. For a color image, each pixel must know the light intensity for each of the three primary colors. Given the intensity information, a picture can be 'painted' pixel by pixel, scan line by scan line.

Whereas the display itself is strictly a hardware problem, the software problem is chiefly this: What intensity should each of the 250,000 odd pixels have in order to create the desired picture? The preceding question assumes a microscopic perspective, whereas the actual software development proceeds at a macroscopic level. The overall software problem divides itself into several major sub-problems, such as spatial orientation (translation and rotation of the object), perspective, hidden surface removal, and reflectivity. This brief overview omits discussion of the solution to these problems, but the reader is referred to a sampling of literature addressing these problems.[1,2,3]

One point must be made here, however. Continuous tone graphics concerns itself with surfaces - specifically surfaces of mathematical models. Consequently, only surface definitions, as opposed to line or point definitions, can be used as input data. One way to define an arbitrary surface is to approximate it as a network of discrete polygonal elements (triangles and quadrilaterals) which are defined first by vertices in 3-D space, and further by a connecting perimeter. Such a definition will herafter be referred as a panel definition.

The continuous tone pictures in this thesis were photographed off a Comtal Image Generator. The display files were generated using MOVIE.BYU - a powerful graphics package written by Dr. Christiansen

---

[1] Henry N. Christiansen, "Applications of Continuous Tone Computer-Generated Images in Structural Mechanics," Structural Mechanics Computer Programs - Surveys, Assessments, and Availability, University Press of Virginia, Charlottesville, Virginia, June 1974, pp. 1003-1015.

[2] Henry N. Christiansen, "MOVIE.BYU - A General Purpose Computer Graphics Display System," Proceedings of the Symposium on Applications of Computer Methods in Engineering, University of Southern California, Los Angeles, August 1977.

[3] William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics (New York: McGraw-Hill, 1973)

(of Brigham Young University) and Dr. Stephenson (now at the University of Arizona). This thesis focuses on generating panel definitions from contour data in a format compatible with the requirements of MOVIE.BYU.

Chapter 3

CONVERTING CONTOURS INTO PANELS

A LIMITED TRIANGULATION ALGORITHM

A contour line can be viewed mathematically as the intersection of an arbitrary surface and a plane. In topography, the plane is generally horizontal at a specified elevation. If the surface is closed, its contour lines will likewise be closed loops. A set of contour lines on evenly spaced parallel planes comprise a contour definition of a surface.

Contour lines of an irregular surface, such as found in nature, do not lend themselves to curve fitting, or other attempts at precise mathematical description. The most convenient numerical description of a contour line is perhaps one where the line is approximated as a string of straight line segments. This digitized contour line offers two pieces of information: nodal coordinates, and connectivity of nodes. Connectivity is implied by the sequence in which the nodes are listed.

Triangulation - the process whereby a panel definition of triangular panels is extracted from a contour definition - is greatly facilitated by observing the connectivity inherent in contour data. That connectivity leads us to explicitly note an obvious rule in triangulation: If two nodes of the same contour are to be defined as nodes of the same triangle, they must neighbor each other on their contour line.

Also, no more than two vertices of any triangle may be recruited from the same contour line (except, of course, in the special case where the entire area enclosed by that contour is to be capped off).

Triangulation is most logically carried on between pairs of adjacent contour lines. Consider this pair of contour loops T (top) and B (bottom).

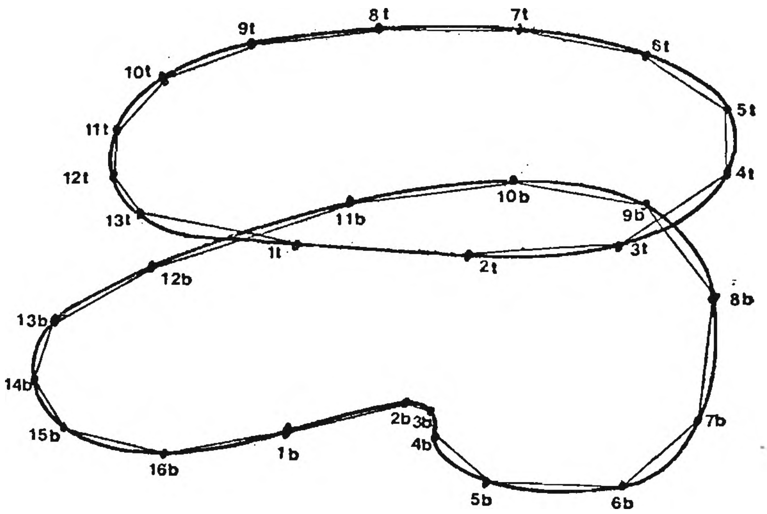


Figure 1

Contour Pair Prior To Triangulation

Two requirements must be met before triangulation commences. First, both loops must run in the same rotational direction, and second, the first nodes of each loop must be proximate. Both rules are met by these loops, and they are ready for triangulation.

Perhaps at this point discussion might best center on the finished product.

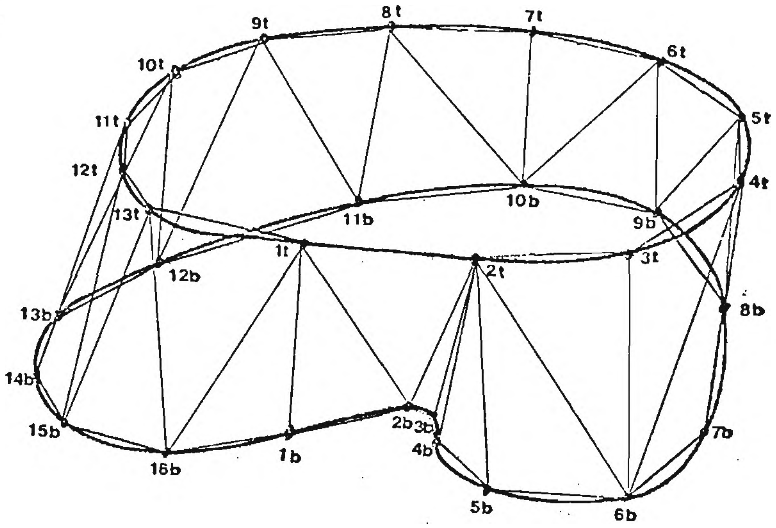


Figure 2

Triangulated Contour Pair

Observe in figure 2 the triangulated contour pair. If one were to ask oneself "How could a computer algorithm be taught to do this?", a few ideas would assert themselves. First, each contour segment can be considered to be the base of a triangle, with the third vertex being a node from the other contour. Secondly, each triangle appears to be as fat as possible. That is, the third vertex is always very near its counterparts on the other contour line.

With these ideas in mind, consider again the untriangulated loops. Referring to figure 3, triangulation commences by defining diagonal 1t-1b. Since contour connectivity requires 1t-2t and 1b-2b as bases of triangles, there are exactly two candidates for the first triangle: 1t-1b-2t, and 1t-1b-2b. Glancing back at the solution,

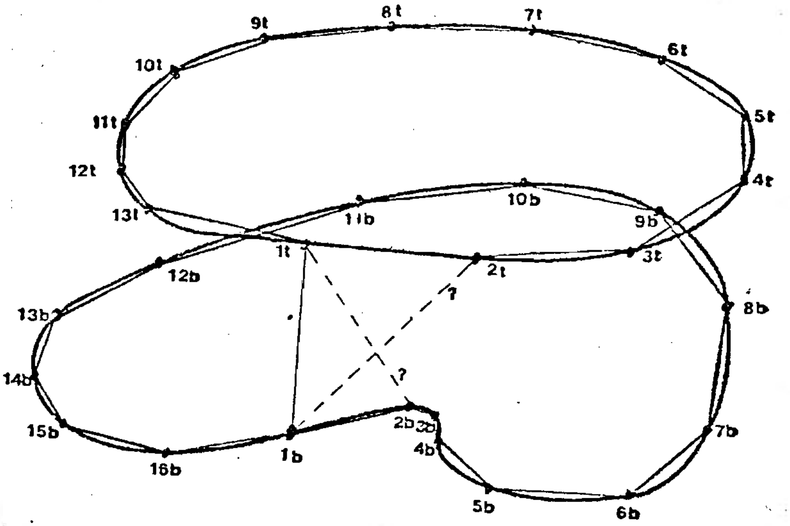


Figure 3

Commencing Triangulation

it is seen that 1t-1b-2b was selected. Moving on, once again there are exactly two possibilities for the second triangle: 1t-2b-2t, and 1t-2b-3b. This time, triangle 1t-2b-2t is selected. Notice that in each case, there are only two triangles to decide between, and that the triangle with the shortest diagonal is chosen. This procedure continues until both loops have been traversed,

This "shortest diagonal" algorithm is very easily implemented, and works fine as long as the two loops are mutually centered and are of reasonably similar size and shape.

MAPPING

The basic "shortest diagonal" algorithm fails for mildly complex cases. A typical example is found in this pair of offset contours.

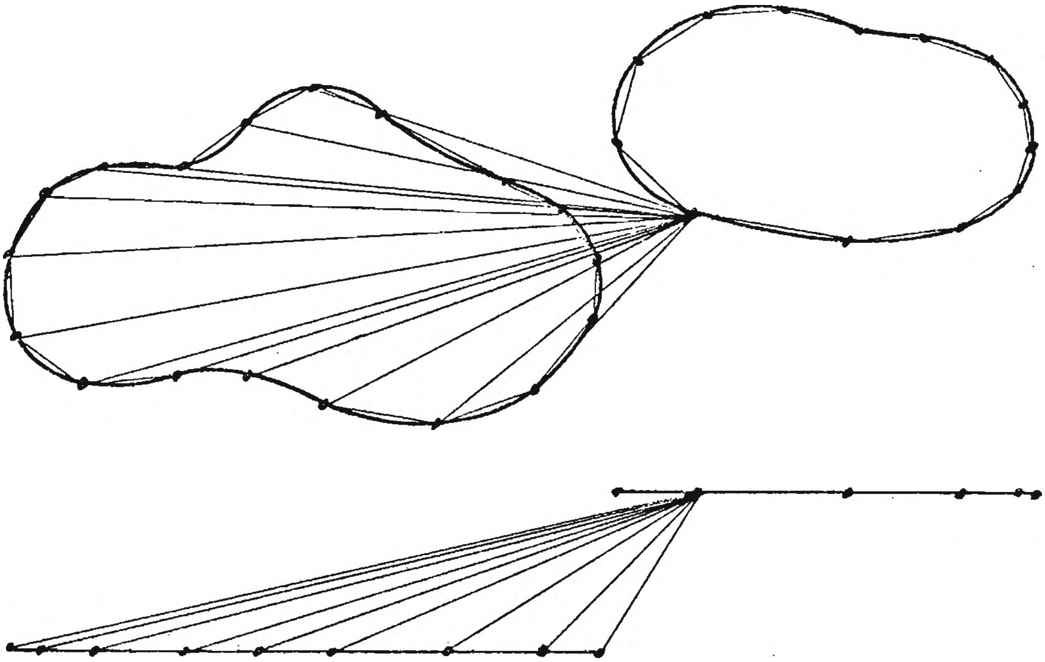


Figure 4

Failure Example

Here, the shortest diagonal search results in a cone. Rather than abandoning the algorithm, let's consider modifying the contour loops to make them more acceptable. As mentioned, the algorithm prefers contour pairs to be mutually centered, of similar size, and of similar shape. The first two requirements can be met by mapping the loops onto a unit square prior to triangulation. (Mapping also tends to make the shapes more uniform, though not always enough. This problem is addressed in the next section.)

Mapping is easily done using translation and scaling functions. Each contour is mapped consecutively in the following manner:

1. Define the rectangular window which encloses the contour.

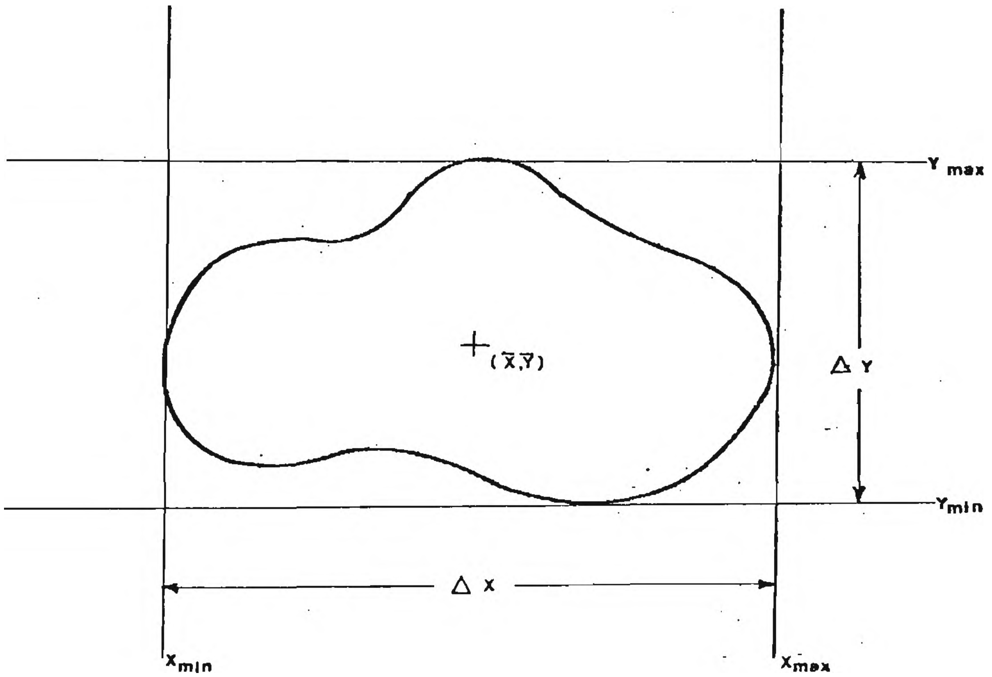


Figure 5

Window Parameters

2. Calculate ΔX, ΔY, $\overline{X}$, and $\overline{Y}$.

3. Map onto a unit square centered at (0,0) by translating and scaling the contour such that its window matches the unit square's window. The equations for this are:

$$X' = (X - \overline{X}) / \Delta X$$

$$Y' = (Y - \overline{Y}) / \Delta Y$$

The mapped contour pair looks like this:

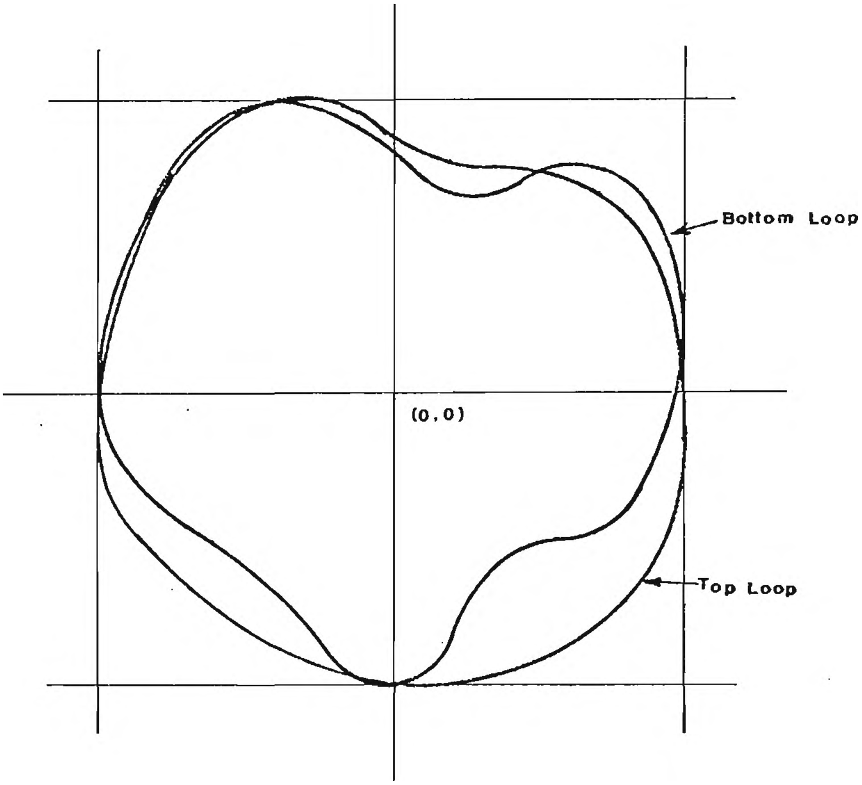


Figure 6

Mapped Contour Pair

With both contours thus mapped, they are easily handled by the original algorithm.

A fringe benefit of mapping is that the resulting triangles tend to align themselves with diagonals that are biased in the direction of the offset. This creates a desirable longitudinal texture.

ULTIMATE AMBIGUITY

A set of contour lines contains the following mathematical information:

1. Exact coordinates of some points on the surface.
2. Approximate gradients in the X-Y contour plane.
3. A general idea of the range of possible Z-gradients.

A panel definition contains items 1 and 2 and improves on item 3 by pinning down approximate Z components of surface gradients. Consequently, there is a degree of ambiguity inherent in the triangulation problem.

When two loops are similarly shaped, the ambiguity is negligible. To illustrate, consider these two solutions of the same triangulation problem;

Figure 7

Synonimous Triangulation Interpretations

Since these two solutions are different, one of them is probably a more exact approximation of the actual surface. But, since the true surface gradients are not available for comparison, and since the two solutions are so similar, either solution is probably adequate. After all, contour lines form a skeletal framework that cast rather rigidly the shape of the surface.

However, as the respective shapes of a contour pair become increasingly divergent, the ambiguity becomes increasingly pronounced. The following convolution provides a good example:

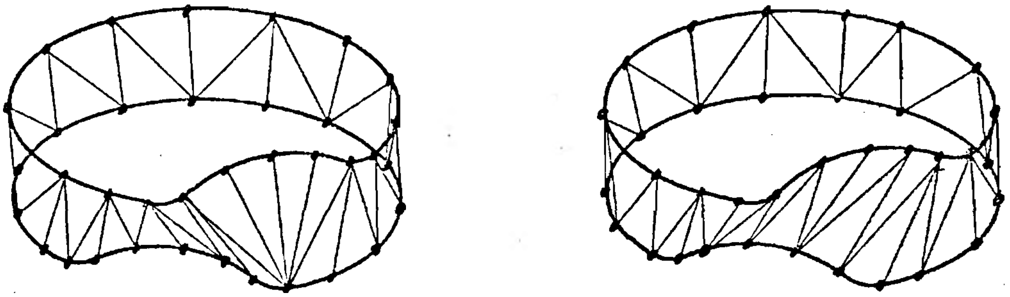


Figure 9

Non-Synonimous Triangulation Interpretations

Here, the variation in interpretation is not as tolerable. Both solutions are reasonable, yet one is wrong. Clearly, more information is required to resolve this problem.

There are two ways to provide the needed information. First, one could require the contour planes to be close enough together that there is minimal variation between adjacent contour lines. This approach has the advantage of tending towards an exact description, and the disadvantage of being uneconomical.

The second approach (adopted in this thesis by default) is to request user interaction to guide the triangulation over cases of excessive ambiguity. This is a more general solution to the problem. Here, the user is called upon to resolve the ambiguity with his knowledge of the true shape of the surface. For the mechanics of how this is implemented in the computer program, refer to the user documentation and the example problem.

BRANCHING

An important feature of this algorithm is the capability to handle branching. Consider this simple case where one contour loop branches into two:

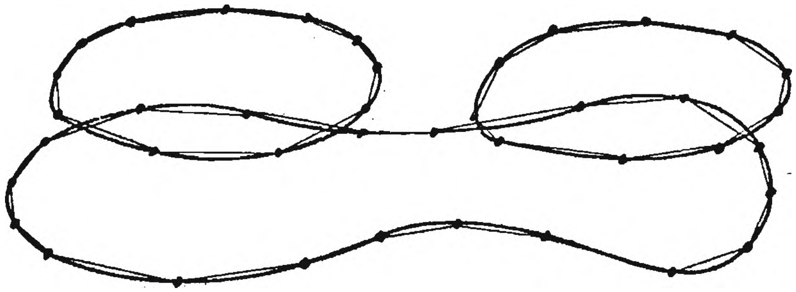


Figure 9

Simple Case of Branching

One way to handle this is to respectively treat each contour as if it were alone, neglecting the other branch. The resulting triangulation would appear like this:

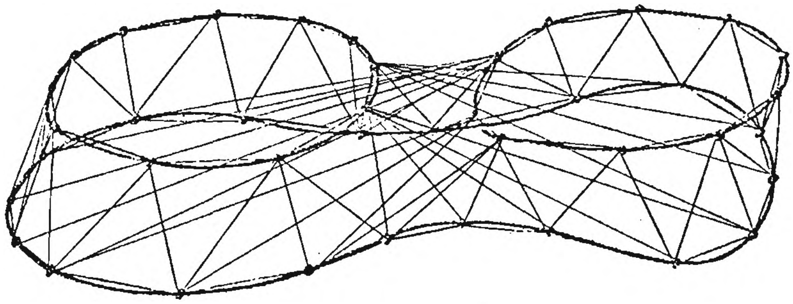


Figure 10

Uneconomical Handling of Branching

Garbled as it looks, hidden surface elimination cleans it up, and provides a smooth transition between branches.

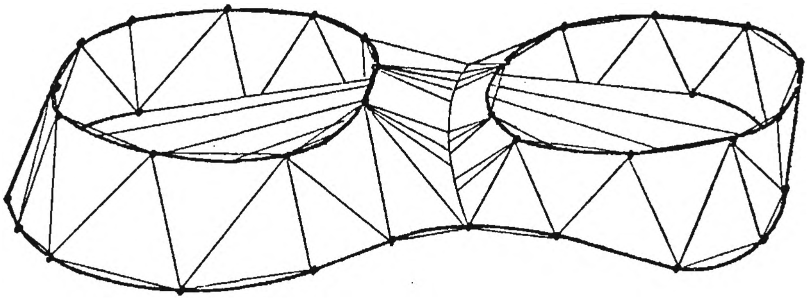


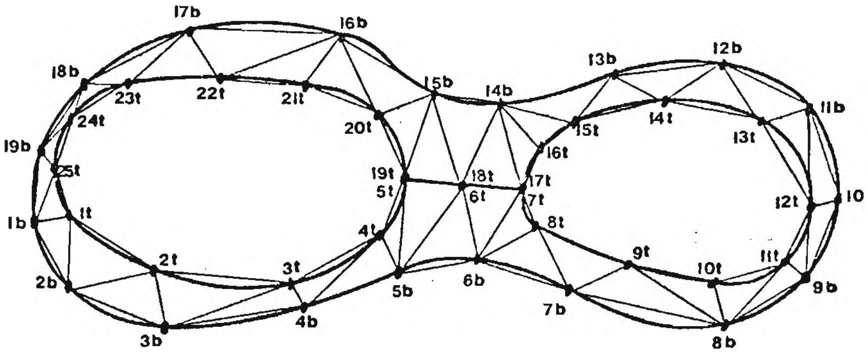
Figure 11

Preceding View With Hidden Surface Elimination

Drawbacks are that it is uneconomical, and it is unacceptable in even mildly complex branching situations.

A more economical, and more general, approach to branching is outlined in this thesis. The idea is to treat all branches as one continuous closed loop by introducing a new node midway between the closest nodes on the branches and renumbering the nodes of the branches and the new node(s) such that they can be considered as being one loop. The Z coordinate of the new node is the average of the Z coordinate of the two levels involved.

Plan



Elevation

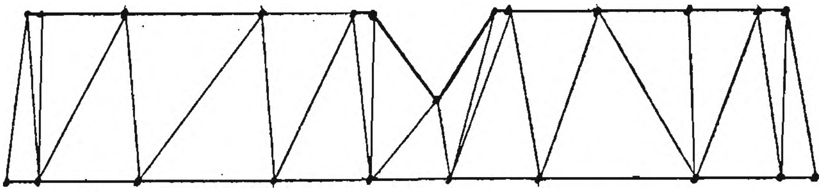


Figure 12

Preferred Handling of Branching

As seen from figure 12, the new node and its immediate neighbors are numbered twice to give the effect of one continuous loop. Triangulation can now proceed as normal. The scheme is easily expanded to handle more than one branch.

Often, there are several contour loops on adjacent planes, posing the problem of loop connectivity. Which loops should be triangulated one-on-one, and which are cases of branching? Judgment, in clear cut cases, can be made on the basis of window overlap.

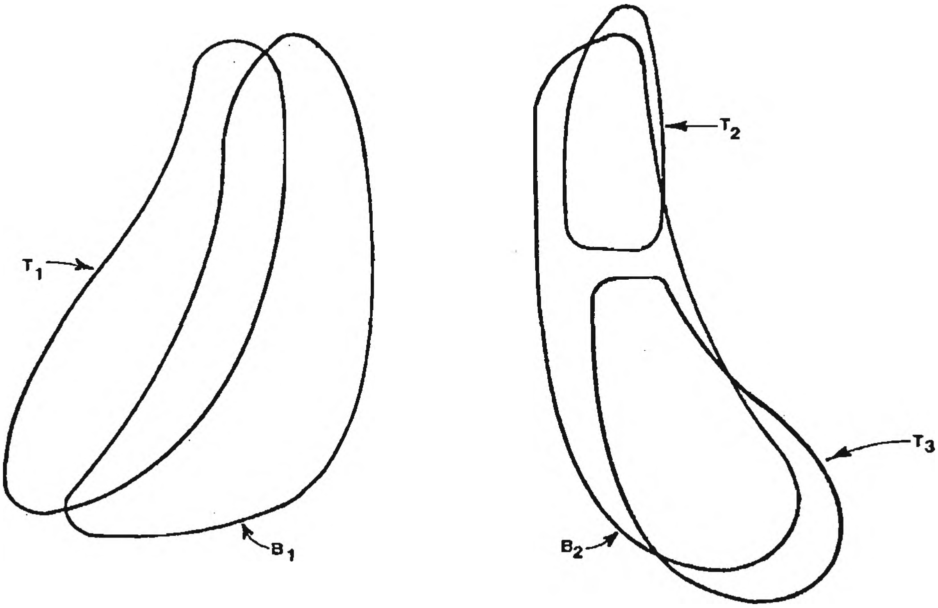


Figure 13

Typical Problem in Connectivity

Here, T_1 and B_1 clearly go together, and B_2 clearly branches into T_2 and T_3. Window overlap is best found by default: IF they don't not overlap, they overlap.

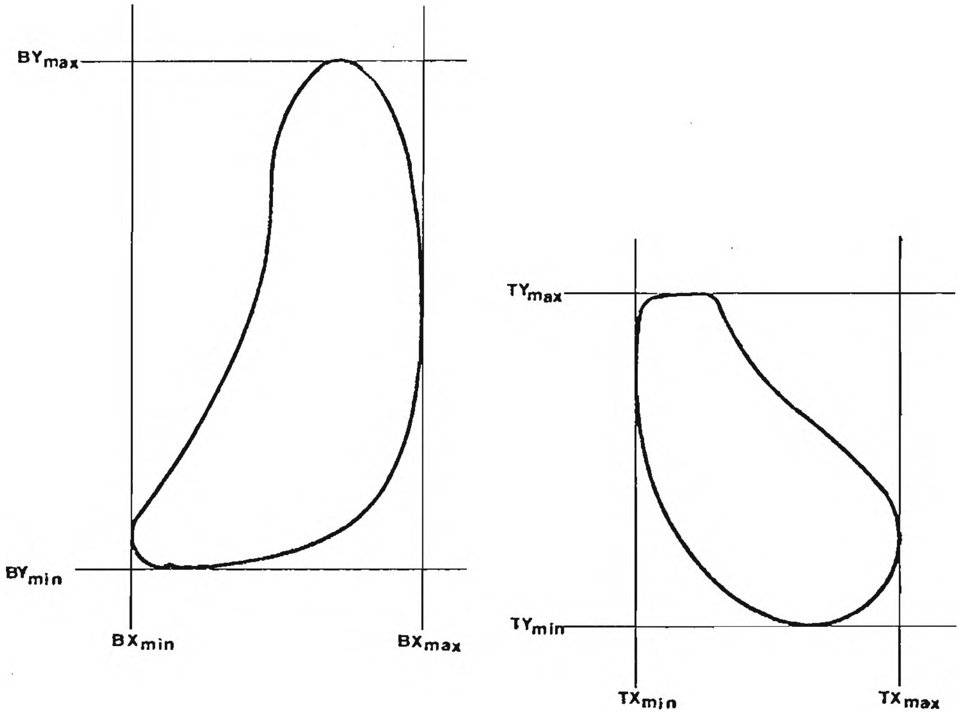


Figure 14

Overlap Test

The rectangular windows definitely do <u>not</u> overlap if:

$TY_{max} < BY_{min}$ or

$TY_{min} > BY_{max}$ or

$TX_{max} < BX_{min}$ or

$TX_{min} > BX_{max}$.

On the other hand, if all four inequalities are false, the windows necessarily overlap.

The algorithm works well for mildly complex cases, with optional user interaction capabilities to handle complex branchings.

Chapter 4

BRAIN CONTOUR DATA

ORIGIN AND DESCRIPTION

The brain data to be used for example purposes in this thesis has an interesting history. In 1967, the first of several movies was made of a human brain at the University of California at San Diego. Using the process of cinemorphology, an entire human brain was placed in a microtome capable of shaving off a slice 25 microns thick. After each slice, a frame of movie film was shot. The entire brain was sliced through, with each successive newly exposed surface recorded on film. Every nth frame of the movie was exploded photographically and outlines traced of each distinct brain structure. Figure 15 shows a cortex contour. In all, 22 separate structures were recorded. The contour outlines were laid on an acoustic tablet and a graduate student (of course!) selected appropriate nodes with the acoustic pen. The nodes were then digitized and recorded.

This digitized data base was accessed by a line drawing graphics package which can produce real time line drawing movies (in color!) on an Evans and Sutherland Picture System.

Each contour plane is referred to as a "page", and there are 98 pages total, ranging from page 3 at the top of the cortex to page 100 at the bottom of the brain stem. The data base is

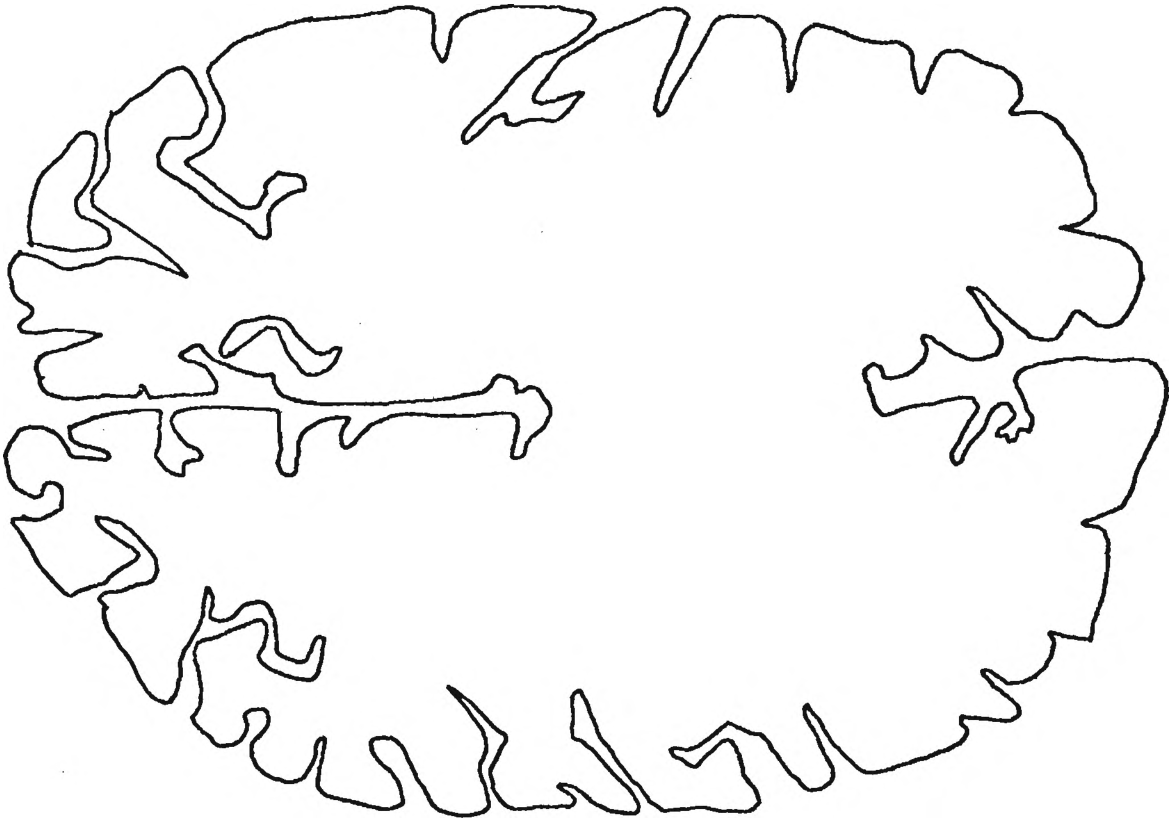


Figure 15

Contour Line of Brain Cortex

massive - totaling 78,651 nodes. Table 1 shows the number of nodes per structure, as well as their page limits. Pages are spaced approximately 1/25" apart, corresponding to a brain that is roughly 4" tall.

Table 1

INDEX TO BRAIN DATA

	STRUCTURE	PAGES	NODES
1	Cortex	3-78	52,870
2	Caudate	30-61	1,922
3	Ventricles	30-84	4,707
4	Fornix	35-57	1,081
5	Putamen	37-54	1,075
6	Thalamus	38-58	1,248
7	Corpus Callosum	41-46	35
8	Globus Pallidus	43-52	725
9	Hippocampus	47-66	1,576
10	Hypothalamus	50-61	400
11	Pineal Body	51-55	92
12	Subthalamic Nucleus	50-56	142
13	Red Nucleus	52-60	238
14	Brain Stem	54-100	1,960
15	Amygdala	55-63	395
16	Substantia Nigra	56-62	243
17	Cerebellum	59-99	6,800
18	Optic Chiasm	60-62	78
19	Mammillary Bodies	57-59	87
20	Mesopallium	19-69	2,385
21	Mammillothalamic Tract	43-56	303
22	Septum	42-49	289

BRAIN DATA FORMAT TRANSMUTATION

The brain contour data arrived at BYU on magnetic tape as 16bit integers in binary format. The data are grouped into 22 structures, which in turn are divided into segments. A segment is a string of contour points and a contour line is formed from one or several segments. Segments represent portions of surfaces which are shared by two structures. Hence, a contour line that is composed of say 5 segments is bordered by 5 neighbors.

Segment definition, which was initially imposed on the data to facilitate line drawing display, somewhat hampers triangulation because all contours must be reconstructed from their constituent segments before triangulation can commence. The problem is aggravated because the segments are randomly sequenced and, furthermore, no convention is observed in clockwise and counterclockwise ordering of nodes.

Segment definition is illustrated by this typical configuration where 3 closed contour loops are defined by 6 segments:

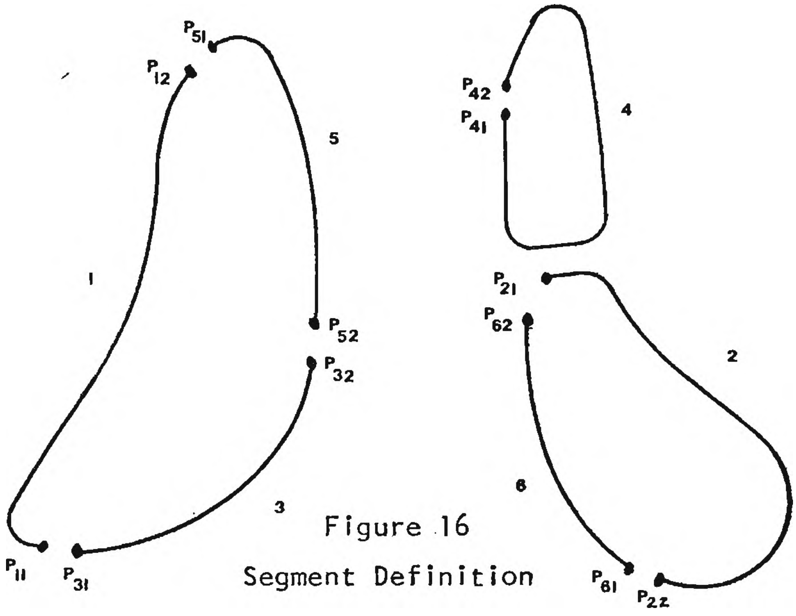


Figure 16
Segment Definition

Notation wise, P_{n1} is the first node of segment n and P_{n2} the last node of segment n. The transmutation algorithm begins by assigning segment 1 to loop 1. A search is made for the nearest neighbor of P_{12} which is P_{51}, and segment 5 is appended to segment 1. Next, the nearest neighbor of P_{52} is sought. Its nearest neighbor is P_{32}. This indicates that segment 3 is sequenced in an order contrary to that of segments 1 and 5. Consequently, segment 3 is joined to loop 1 in reverse order. Since P_{31} neighbors P_{11}, the loop is complete. A flag is set for loops 1, 3 and 5 preventing future assignment. This logic repeats until all segments are joined to a loop.

It is important to impose the convention that loops run uniformly in a clockwise (or counter-clockwise) direction. To enforce this convention, all nodal angles of a contour line are summed for monitoring in the following manner:

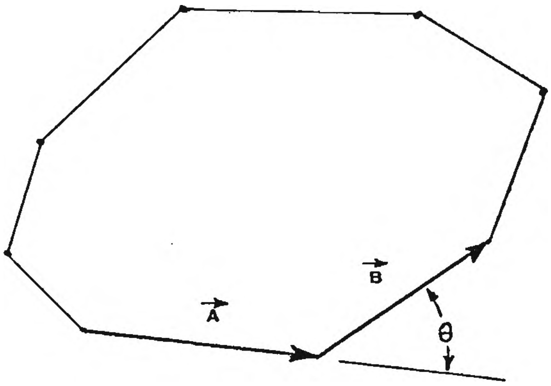


Figure 17

Determining Direction of Rotation

Theta is the angle by which each succedent vector deviates from a straight line. The sum of all such angles will be 360 degrees for counterclockwise sequencing and -360 degrees for clockwise. Theta is computed from vector cross and dot products.

$$\sin\theta = \frac{\vec{A}\times\vec{B}}{AB}$$

$$\cos\theta = \frac{\vec{A}\cdot\vec{B}}{AB}$$

$$\theta = \begin{cases} \sin^{-1}(\sin\theta) & , \cos\theta>0. \\ \sin^{-1}(\sin\theta)+90 & , \cos\theta<0 \text{ and } \sin\theta>0. \\ \sin^{-1}(\sin\theta)-90 & , \cos\theta<0 \text{ and } \sin\theta<0. \end{cases}$$

In a few instances, the brain data invalidates this approach by having a contour line cross itself like this:

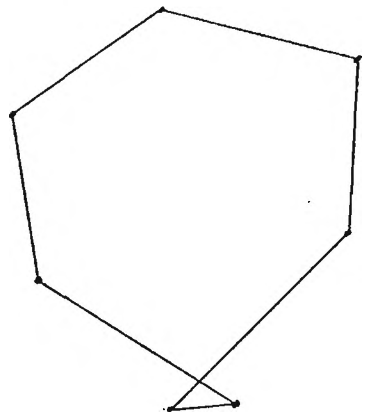


Figure 18

Brain Contour Error

This error causes a figure 8 which results in $\Sigma\theta$ approaching 0. Often, this causes a violation of the rotation convention.

Chapter 5

ECONOMIZING

NODE ELIMINATION

If a data base is too refined (i.e. contains nodes you could do without) it is desirable for reasons of economy to eliminate the less essential nodes. Consider this node:

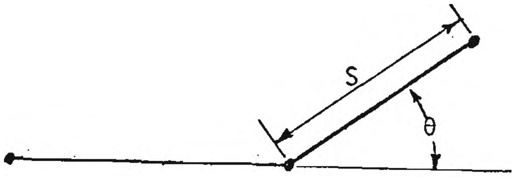


Figure 19

Node Elimination Parameters

The node is accepted (or rejected) upon the following criteria:

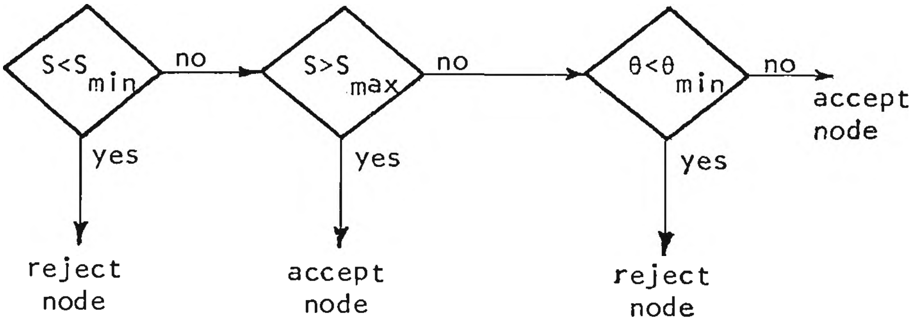


Figure 20

Node Reduction Flow Chart

S_{min}, S_{max} and θ_{min} are user definable parameters. Every node is screened using this logic. To assure acceptance of every node, all three parameters may be set to zero.

This is a logical place to interject a few thoughts on interpolation of new nodes. Since a digitized contour line is an approximation comprised of a series of straight line segments, it is reasonable to assume, and important to prescribe, that nodes are selected such that the digitized approximation does not deviate intolerably from the actual contour line. This would imply a correlation between nodal density and contour line curvature. That correlation suggests that it is desirable to re-distribute nodes around the contour loop according to a curvature vs. node density function using curve fitting procedures. This is an appealing thought, since it would reduce angularity in the continuous tone display. This would be great, provided the actual surface isn't angular. Of course, however, that assumption is not always valid. Take, for example, a simple four node contour definition of a square. If curve fitting were imposed in an attempt to extrapolate extra nodes, the result would tend towards a circle. Angularity is reduced at the expense of accuracy. In conclusion, the burden of providing acceptable data rests with the person who actually does the digitizing.

QUADRILATERAL FORMATION

For economy of storage, it is desirable to join pairs of adjacent triangles together into quadrilaterals. Not all ajoining triangles are thus combined. The decision is made on the basis of how warped the resulting quadrilateral would be, that is, the angle by which the two triangles are out of plane. That angle is easily found using vector algebra.

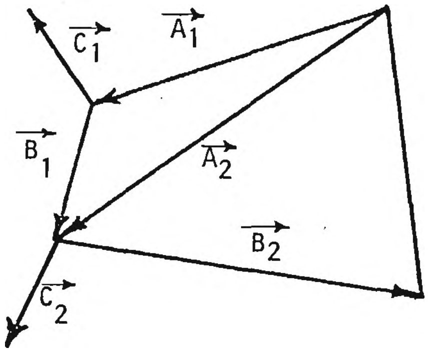


Figure 21

Warp Angle Determination

$$\vec{C}=\vec{A}\times\vec{B}$$

$$\sin\alpha=\frac{\vec{C}_1\cdot\vec{C}_2}{C_1\ C_2}$$

If $\sin\alpha<\sin\alpha_{max}$, the two triangles are redefined as a quadrilateral. α_{max} is user definable and defaults to 45 degrees.

Chapter 6

FORTRAN IMPLIMENTATION

A few explanatory remarks are offered here to the reader who is bent on deciphering the source code.

As mentioned, the brain data arrived at Brigham Young University in the form of 16 bit integers on tape in binary format. To facilitate use on the DEC-10 computer, these data were re-formatted into 7 bit ASCII data files, one file for each of the 22 structures, with 8 integers per line. The first two integers of a file comprise the "structure heading", the first integer being the structure number, and the second being the number of segments in the structure. These two integers are ignored by the triangulation program.

The next four integers form the segment header of the first segment. The first integer of the segment header is the page number or horizontal level of the segment. The page numbers range from -3 to -99. The Z coordinate of the segment is computed from the formula RZ=-(Z+51)*450*SCALE where RZ is the Z coordinate, Z is the page number, and SCALE is the scale factor. The second and third integers in the segment header are ignored. The fourth integer, NPL, is the number of nodes in the segment.

Immediately following the segment header are the X-Y coordinates of the segment nodes, totalling NPL pairs. The next segment header immediately follows the last node of the preceding segment, so there is an uninterrupted string of integers from start to finish in the data file. Nodes of all segments of the same page are stored in array P2. The segments are joined together to create closed loops (as described in chapter 4) and stored in array P3. Node elimination is imposed, and the nodes are finally stored for triangulation in array P.

Two pointers are used in accessing the nodes in array P. The Loop Pointer - LPP - indexes the global loop numbers of the first loop on any contour level. Pointer P1 addresses the global node number of the first node on a loop. For example, the X coordinate of the n^{th} node of the j^{th} loop on the k^{th} contour level would be:

$$P((P1(LPP(k)+(j-1))+n-1)),1)$$

Knowing these conventions, the fortran coding should be relatively lucid. The coding, written for use on a DEC-10 system, is given in the appendix. An additional assembly language Tektronix interface, GRTEK.REL, must also be loaded.

Chapter 7

EXAMPLE PROBLEM

A simple yet dramatic example of branching is provided by the caudates - the symmetric pair of brain structures shown below.

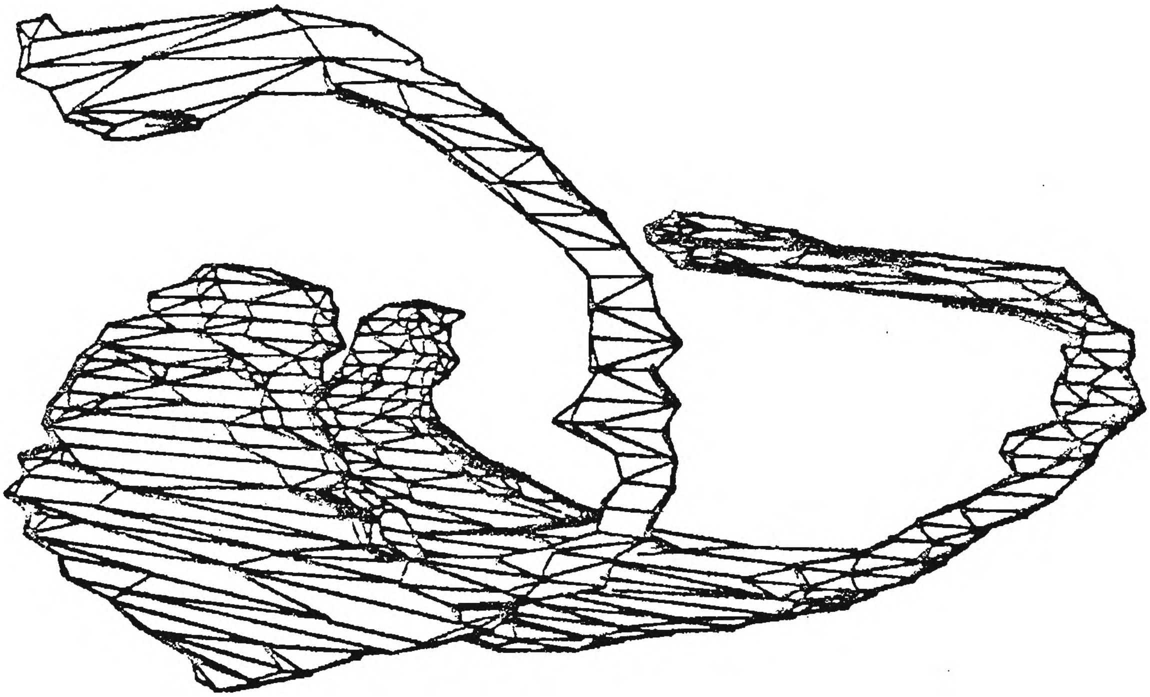
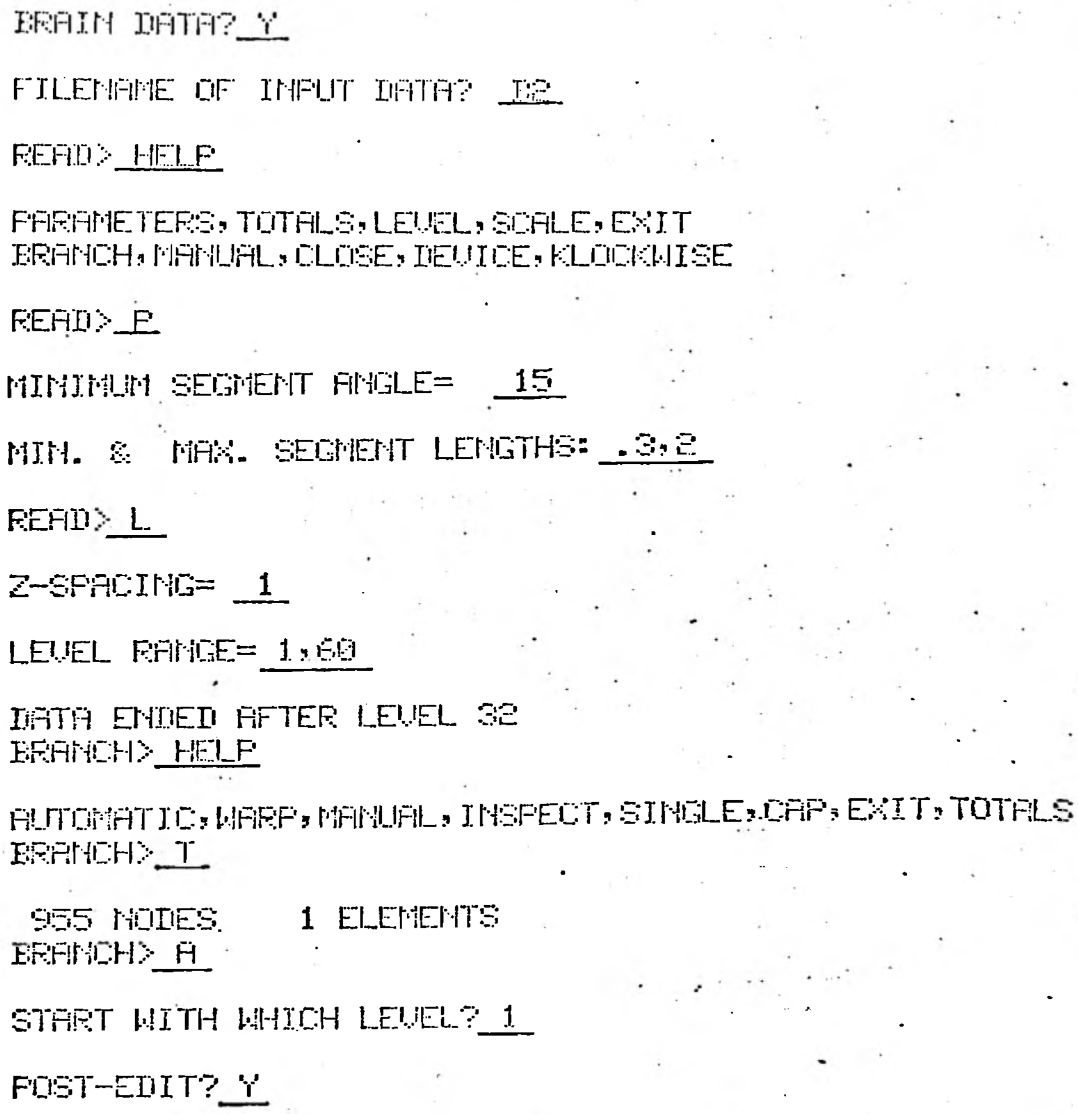


Figure 22

Caudates

Triangulation of the caudates occurred as follows. User statements in the computer dialogue have been underlined to distinguish them from computer generated prompts. Refering to figure 23 the computer begins by asking if the data it is about to read is brain data. It is, and the caudate data file name, B2, is given. Next, the menu of commands is printed, of which PARAMENTERS is



```
BRAIN DATA? Y

FILENAME OF INPUT DATA?  D2

READ> HELP

PARAMETERS,TOTALS,LEVEL,SCALE,EXIT
BRANCH,MANUAL,CLOSE,DEVICE,KLOCKWISE

READ> P

MINIMUM SEGMENT ANGLE=   15

MIN. &  MAX. SEGMENT LENGTHS: .3,2

READ> L

Z-SPACING= 1

LEVEL RANGE= 1,60

DATA ENDED AFTER LEVEL 32
BRANCH> HELP

AUTOMATIC,WARP,MANUAL,INSPECT,SINGLE,CAP,EXIT,TOTALS
BRANCH> T

 955 NODES     1 ELEMENTS
BRANCH> A

START WITH WHICH LEVEL? 1

POST-EDIT? Y
```

Figure 23

Computer Dialogue

selected. We opt to set the minimum segment angle to 15 degrees, S_{min} to .3 and S_{max} to 2. (These parameters are explained on page 26). Having returned to the READ> prompt, we now choose to read in the data, being satisfied with the default values for SCALE, CLOSE, and KLOCKWISE. After setting the LEVEL parameters, the computer goes to work reading in contour segments, reconstructing them into loops, thinning them out, and assuring that all loops run in a clockwise direction. The algorithm encounters end of file before 60 levels are read in, and informs us that all 32 available levels have been read in and processed. The TOTALS command receives the response that there are 955 nodes in array P, which means that over half of the available 1922 nodes have been thinned out by the node elimination algorithm.

The Caudates are quite regular, requiring little or no interaction to triangulate properly, so the AUTOMATIC option is invoked beginning, as usual, with level 1. As a safeguard, post-editing is requested. If the data were unquestionably obedient, the post-editing could justifiably be circumvented, but this way good results are guaranteed.

The algorithm now proceeds to first determine connectivity, then to triangulate all loops implicated in the the window overlap connectivity check, and finally to display the resulting panel definition. The user glances at each successive display and grants acceptance with a carriage return, or occassionally rejects the triangulation, as the case may be. (Usually it is immediately clear when triangulation is unacceptable. Normally, failure occurs in areas where the two loops are excessively dissimilar, and the resulting panels are often bizzare).

Two of several simple one-on-one triangulations are shown in figures 25 and 26 , each of which is accepted. However, the branching loops in figure 27 are triangulated incorrectly, and a change is requested. This change is granted through erasure of the screen, re-drawing of the untriangulated loops, and issuing of the TRIANGULATE> prompt. The INTERACTIVE command is given. As always, only the first letter of the command is required. Referring to figure 28 the nodes are numbered for identification - even for the top loop and odd for the bottom.

The INTERACTIVE command allows the triangulation to be controlled by allowing specification of nodal delimiters between which triangulation will occur. For example, if the delimiteres 1,1 for the top and 1,2 for the bottom were chosen, only one triangle would be formed. Basically the selection of delimiters is a trial and error process. The user delimits as large a span as reasonable. If the resulting triangulation is adequate, great! If not, try again with a smaller span. A general rule might be to procede quickly over sections where the two contours are similar, and cautiously over sections where they are dissimilar.

Figure 24

Continuous Tone Rendering
of Caudates

Referring to figure 26, the triangulation was successful through node 17 of the bottom loop, but then failed to traverse the branch properly. This suggests the selection of delimiters 1,28 for the top and 1,17 for the bottom. The resulting triangulation shown in figure 28 looks good. Now, one more span (top:28,33 ; and bottom:17,20) should suffice, and figure 29 confirms the hope. Now that the entire circuit is complete, AUTOMATIC mode is re-entered and triangulation proceeds smoothly to the conclusion. Upon exiting from the program, the panel definition is written into a disk file, available for display using MOVIE.BYU. A continuous tone image created by that panel definition is shown in figure 24.

CHANGES? NO

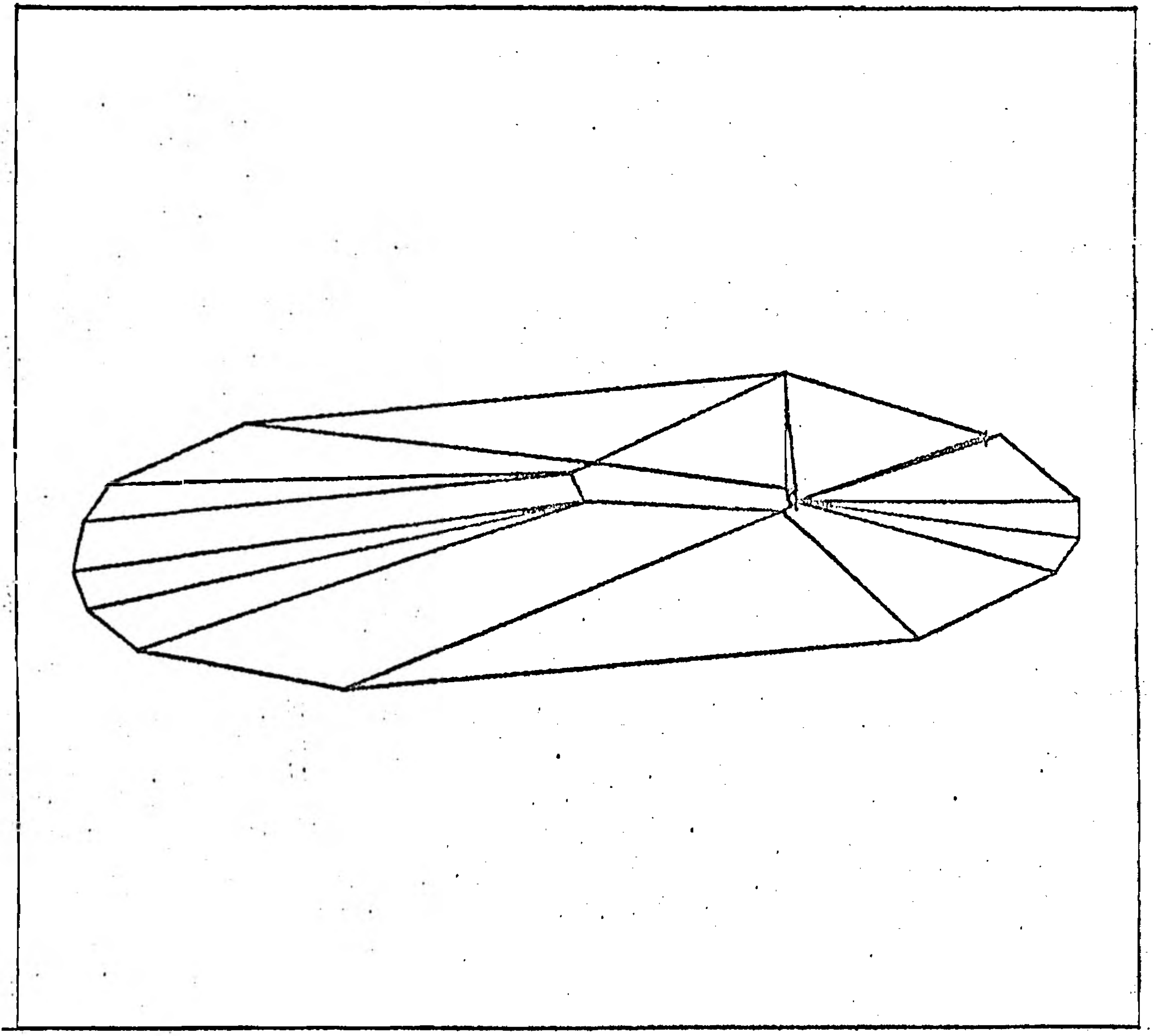


Figure 25

Example

CHANGES? <u>NO</u>

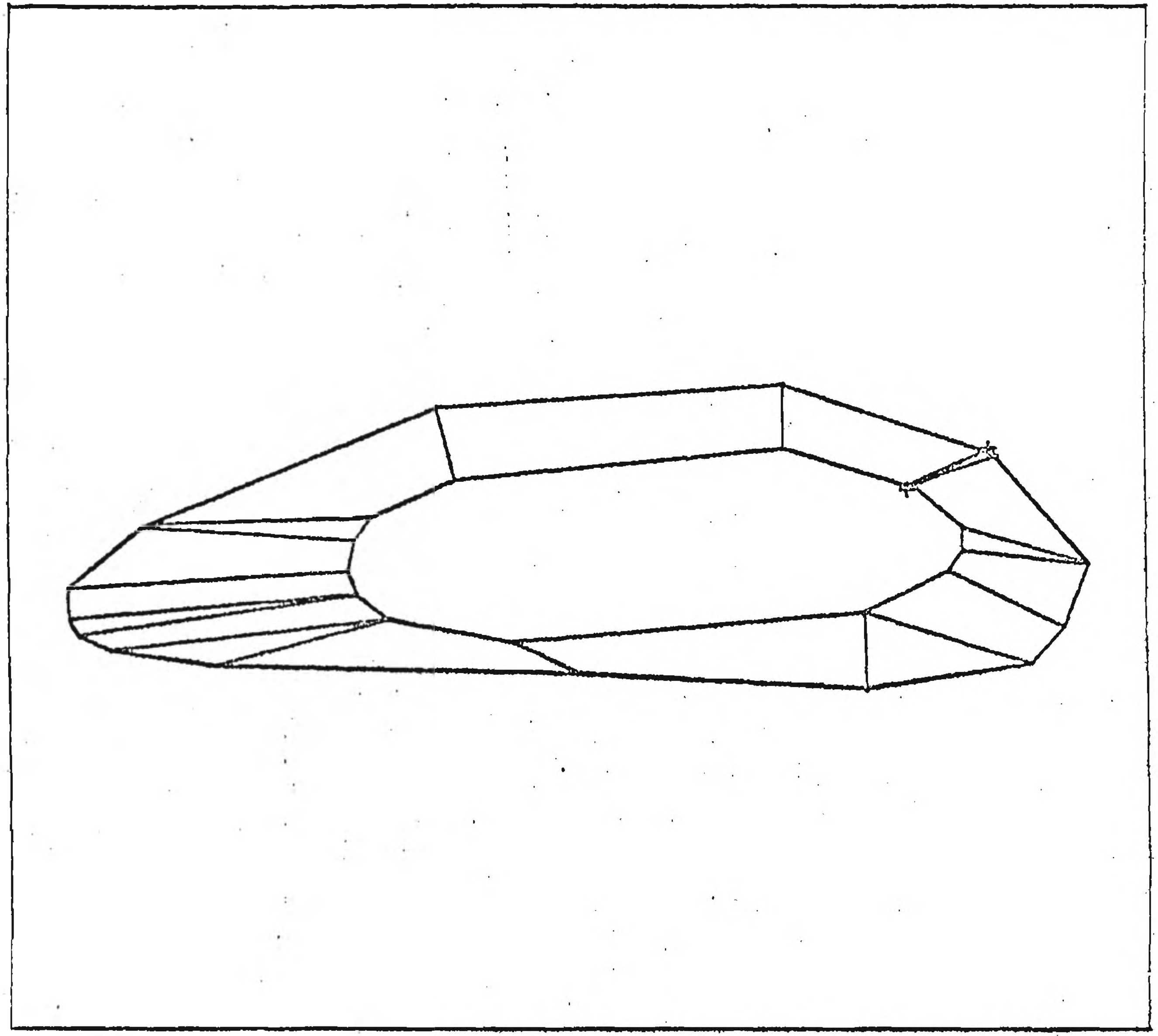


Figure 26

Example

CHANGES? YES

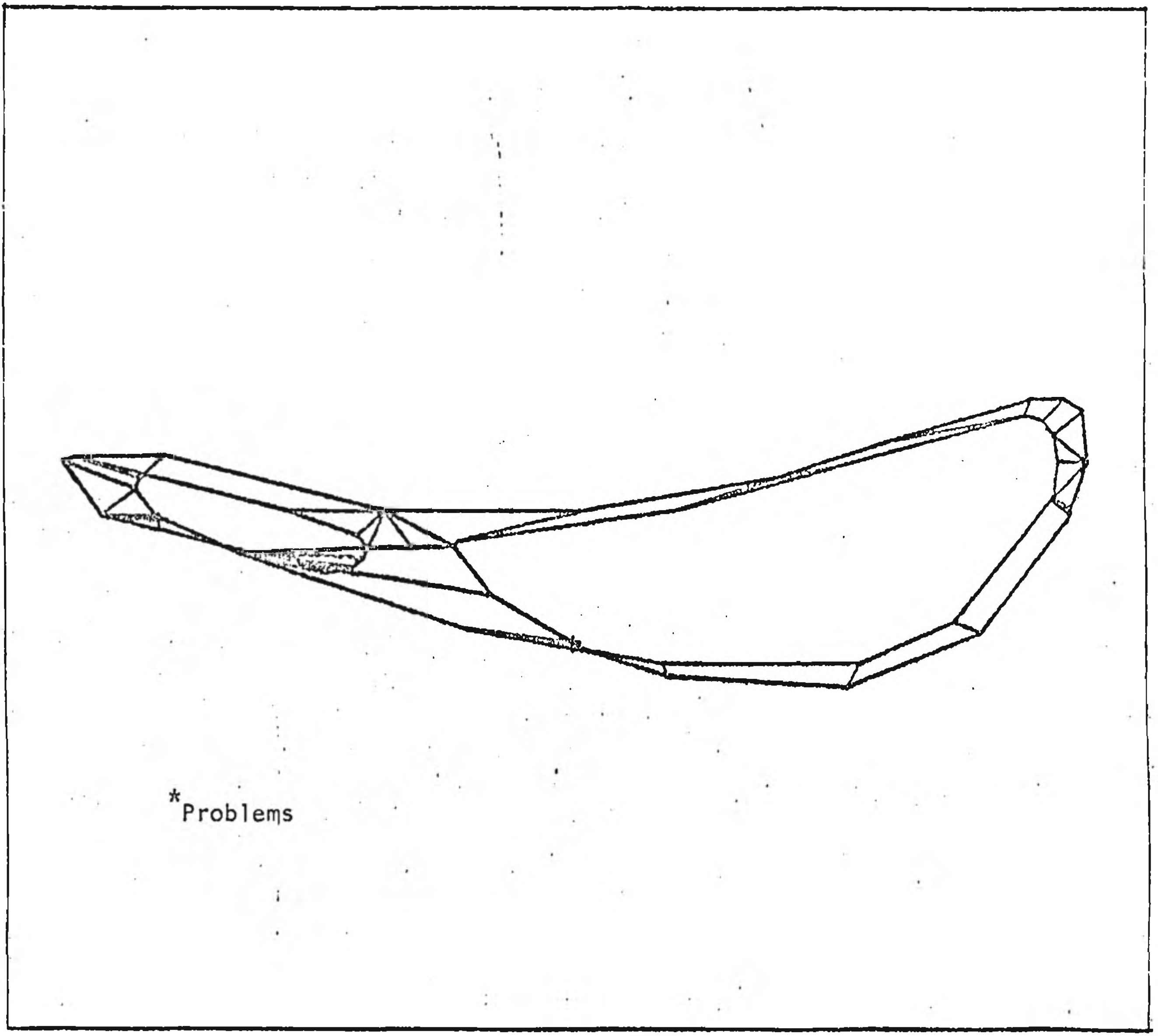


Figure 27

Example

```
TRIANGULATE>1

TOP:   1-33
1,28

BOTTOM:  1-20
1,17

CHANGES? NO
```

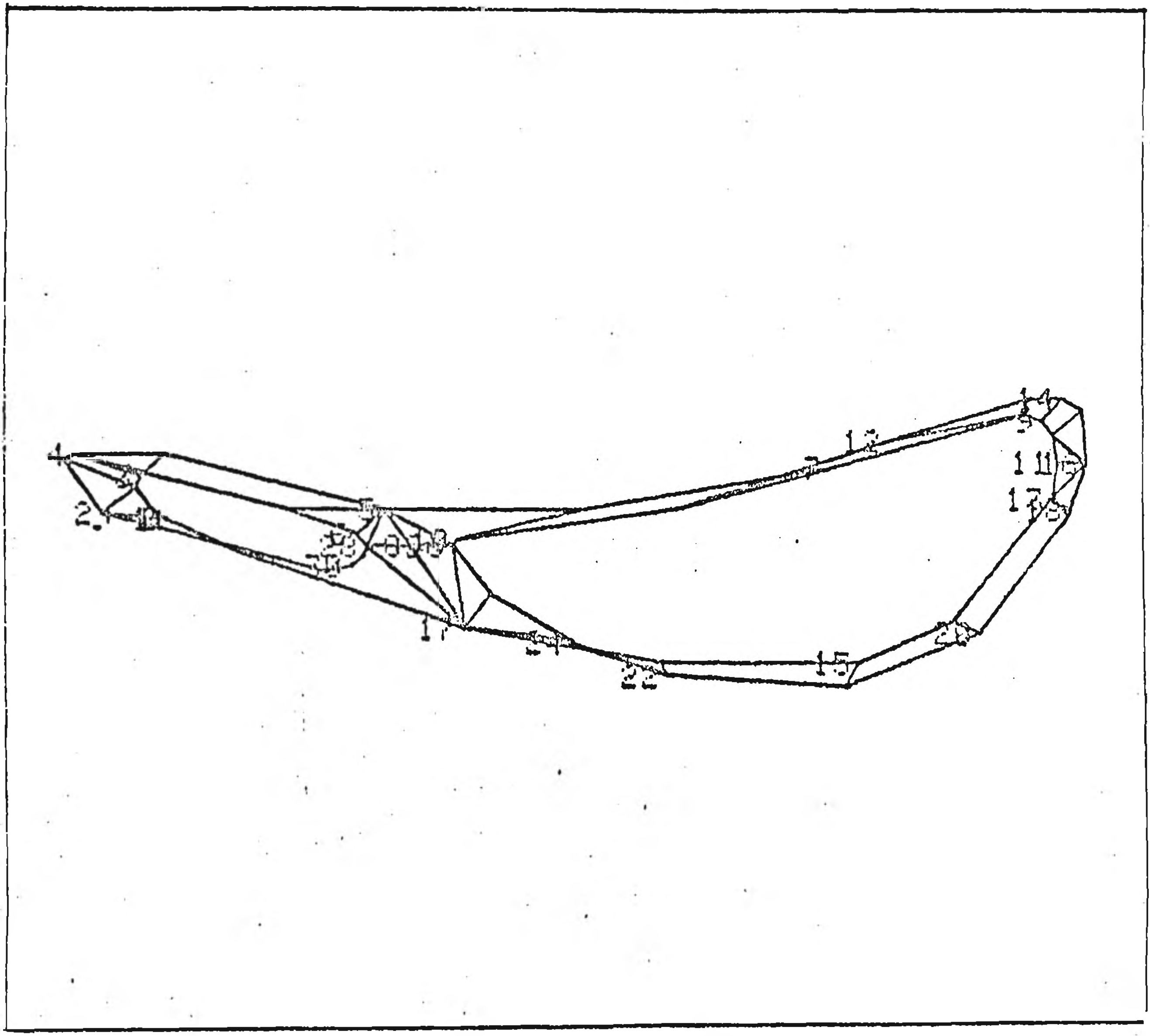


Figure 28

Example

TOP: 28-33
<u>28,33</u>

BOTTOM: 17-20
<u>17,20</u>

CHANGES? <u>NO</u>

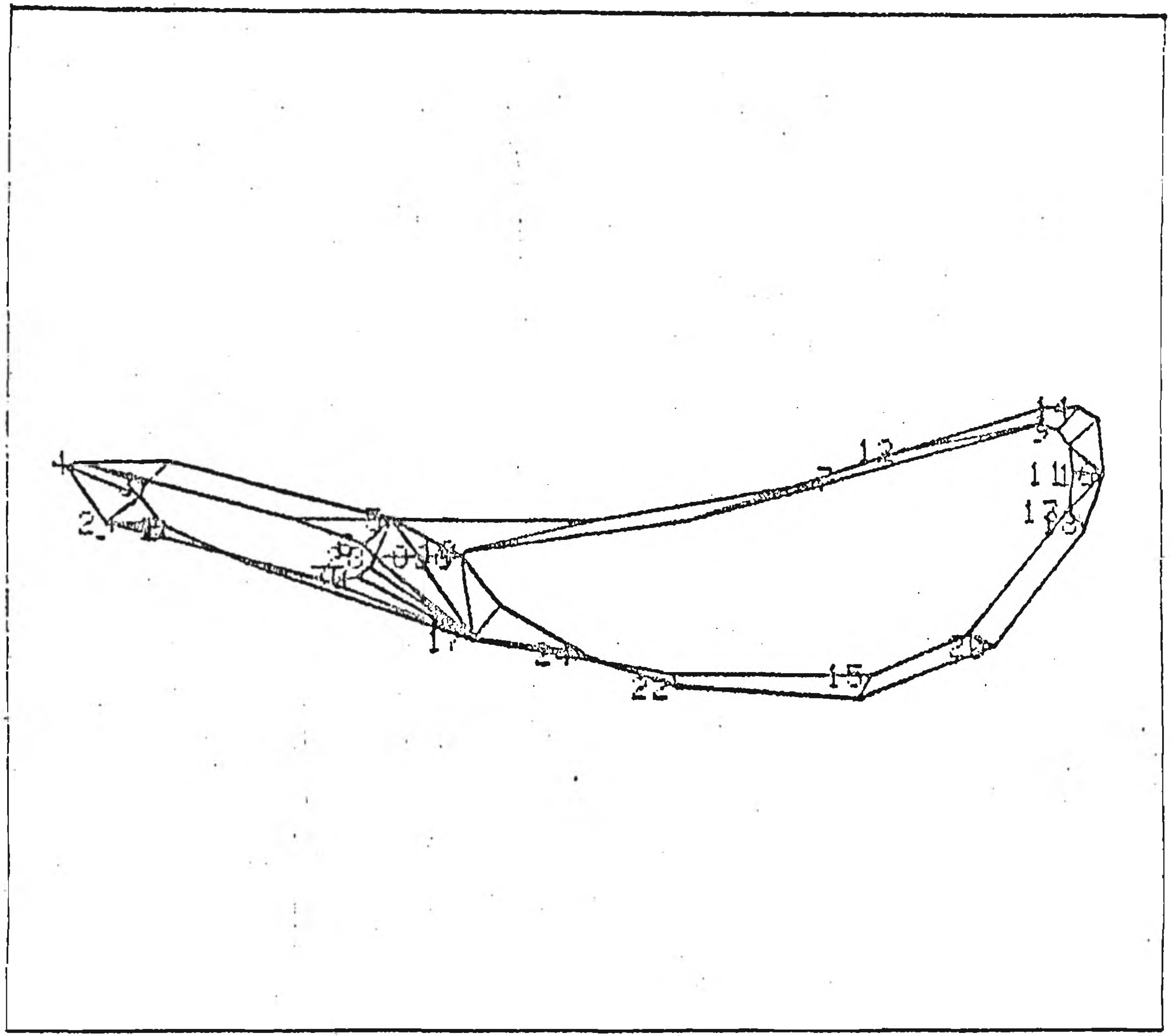


Figure 29

Example

Chapter 8

PICTURES

This chapter presents examples of the finished product. It is difficult to judge how true to life the images are, due to the highly esoteric nature of the subject matter. Nonetheless, it is generally evident that the triangulation algorithm has performed reasonably.

The first structure presented is the Brain Stem. The Brain Stem was a straighforward triangulation problem. There are no branches, and there is no sreious variation in the shape of its respective contours. The only difficulty was a case of illeagle rotational direction, as described on page 25. Other than that, the entire triangulation was handled automatically. Two panel definitions were generated for the brain stem. The first file was made with $S_{min}=S_{max}=\theta_{min}=0$, Z-SPACING=1, and LEVEL RANGE =5,36. The resulting panel definition had 1609 nodes and 1991 panels. The second brain stem panel definition has 378 nodes and 492 panels. It was generated with $S_{min}=.2$, $S_{max}=1$, $\theta_{min}=15$, Z-SPACING =4, and LEVEL RANGE =5,40. Line drawings are shown in figures 30 and 31, and continuous tone images in figures 34 and 35.

The next pair of images, figures 36 and 37 are detailed studies of the thalamus. Here, 4 loops branch into 2, then 2 into 1. Figure 37 has a more biological look due to Gouraud smooth surface simulation.

Figure 38 is a striking composition of 6 different structures, each in proper relative orientation. The sructures are identified in figure 32 and figure 33 shows a line drawing. To enable so many parts, the larger structures have coarser panel definitions.

All of the preceding examples were triangulated with virtually no user interaction. The cortex slice, in figure 39 was not so oblidging. This image is presented to demonstrate the degree of complexity the algorithm can accomodate. To help orient the reader, this image represents about a ½" thick slice of the cortex, centered about ½" from the top of the brain. The many oddly shaped holes are due to the fact that the top most convolutions are decapitated in this view. This panel definition - shown here with smooth shading - consists of 1752 nodes with 1778 panels. This particular data did not cooperate with the automatic algorithm, and required nearly 2 hours time to interactively triangulate. The difficulty was not so much the complexity of the shapes, but the dissimilarity between adjacent contours. Also, the data had a disproportionate number of glitches. Nonetheless, the continuous tone image is quite convincing.

One final image is the panoramic shot of Mt. Timpanogos as it might be seen from an airplane flying to the west of Timp. This is submitted to illustrate a possible application of triangulation to display of topographic surfaces.

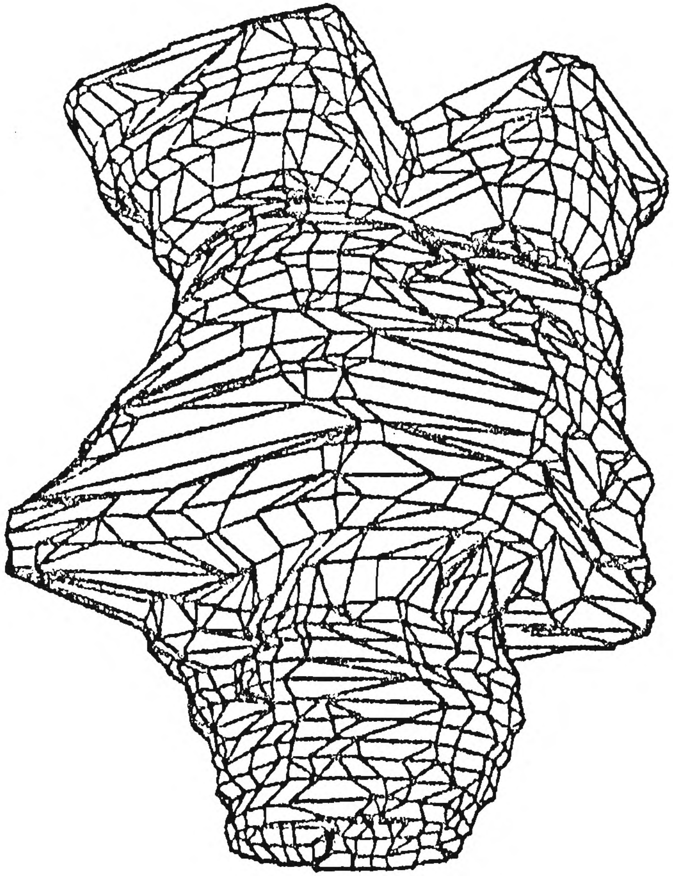


Figure 30

Brain Stem - 1991 Panels

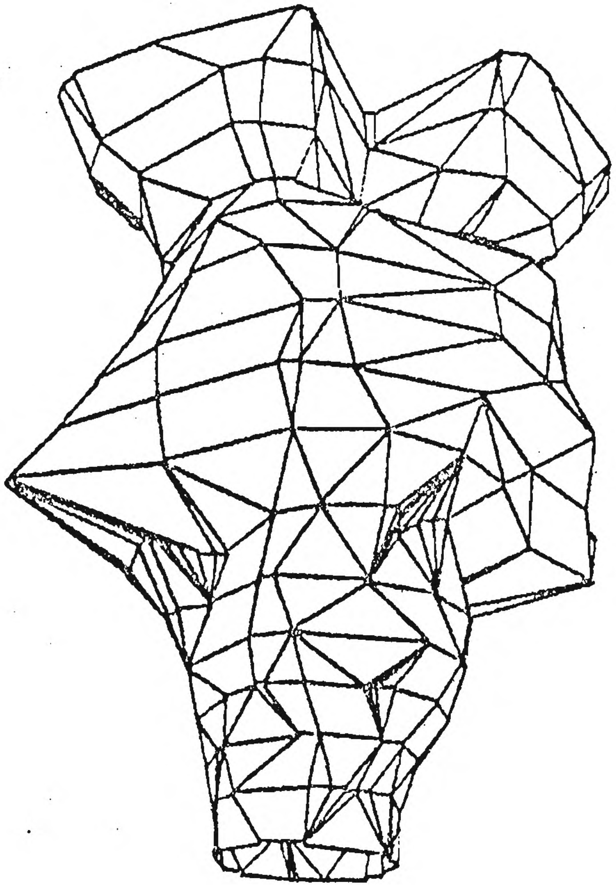


Figure 31

Brain Stem - 492 Panels

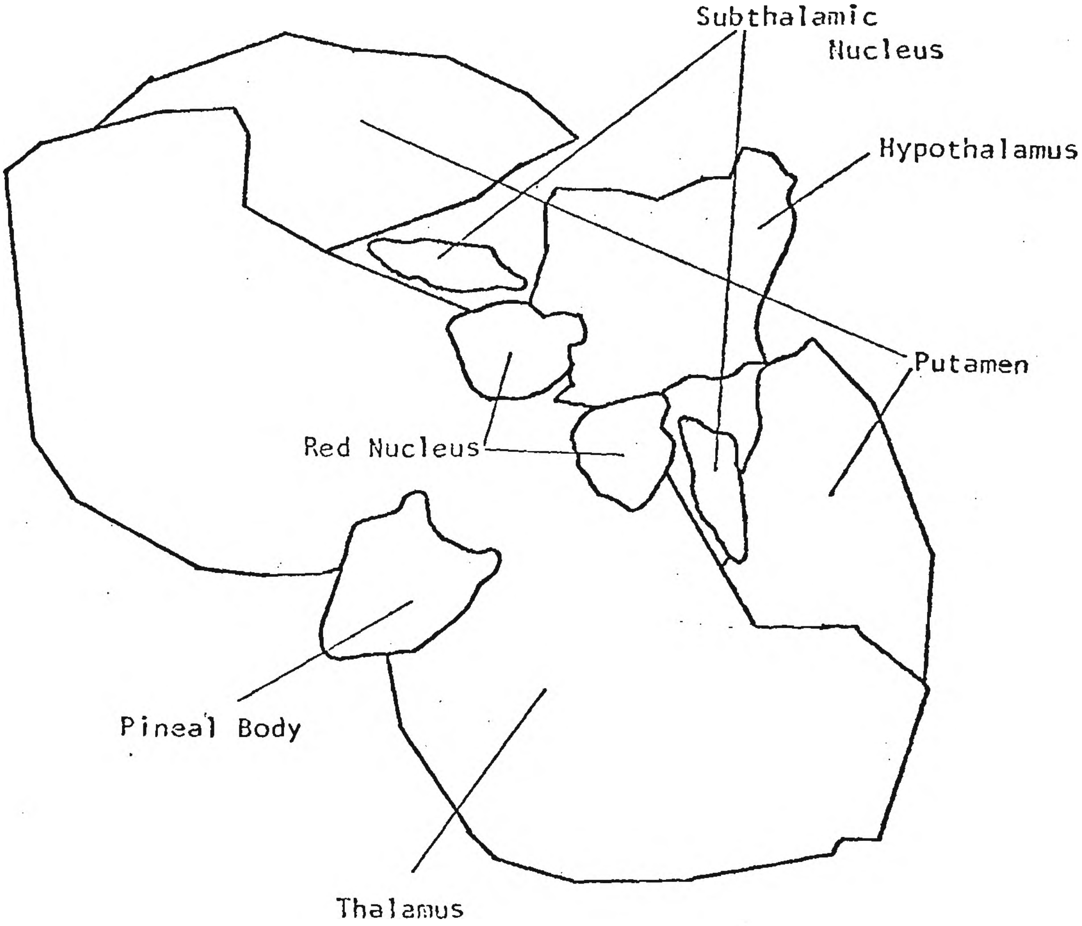


Figure 32

Labelled Composite View

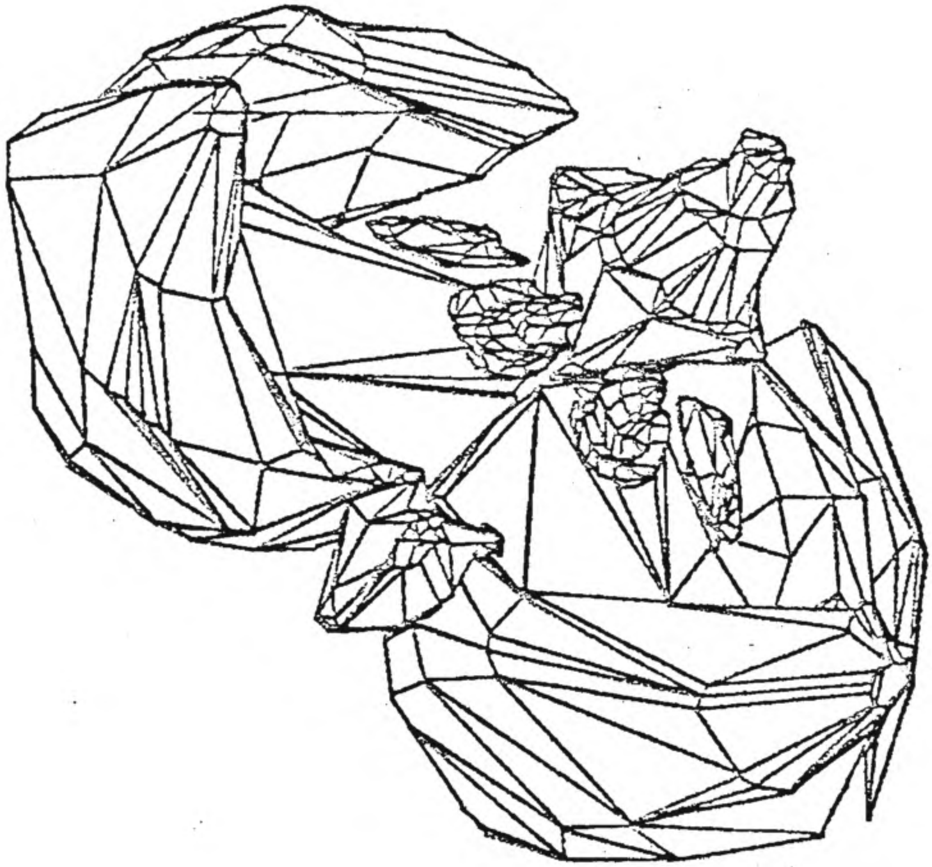


Figure 33

Composite View
Line Drawing

Figure 34

Brain Stem with
1991 Panels

Figure 35

Brain Stem with
492 Panels

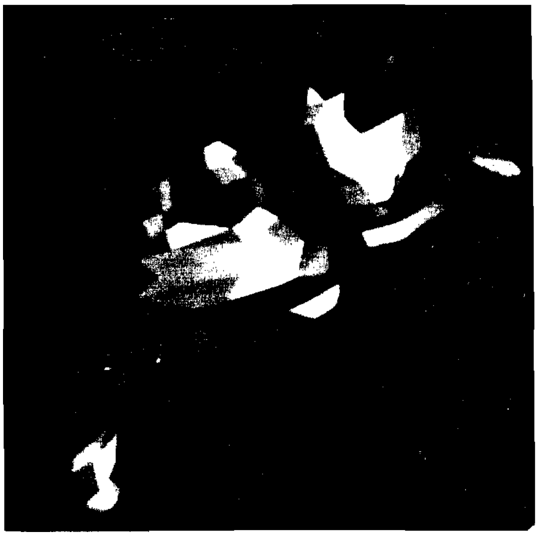


Figure 36

Thalamus with
Flat Shading

Figure 37

Thalamus with
Smooth Shading

Figure 38

Composite View

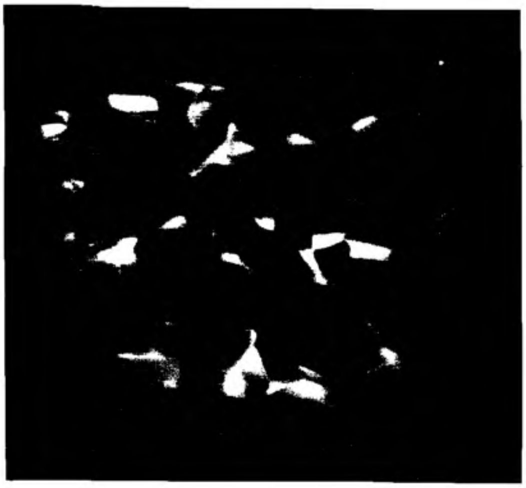


Figure 39

Cortex Slice

Figure 40

Mount Timpanogos - Flat Shading

Figure 41

Mount Timpanogos - Smooth Shading

Chapter 9

CONCLUSIONS

This thesis purports to present a general solution to the problem of converting a contour definition of an arbitrary surface into a panel definition. That assertion is rigorously tested by the brain data, and experience with that highly complex data base lends credence to the claim of a general solution, Total user interaction capabilities virtually gaurantee a general algorithm.

Work might be done on reducing the amount of user dependence in the algorithm, though most reasonable cases require no interaction at all. Also, it would be helpful to improve graphical interaction by using, for example, a tablet to input interaction parameters. Study might also be made on how the economy parameters (S_{min}, S_{max}, θ_{min}, and α_{max}) effect the continuous tone image.

BIBLIOGRAPHY

Christiansen, Henry N. "Applications of Continuous Tone Computer-Generated Images in Structural Mechanics," Structural Mechanics Computer Programs - Surveys, Assessments, and Availability, University Press of Virginia, Charlottesville,Virginia, June 1974, pp. 1003-1015.

________. "MOVIE.BYU - A General Purpose Computer Graphics Display System," Proceedings of the Symposium on Applications of Computer Methods in Engineering, University of Southern California, Los Angeles, August 1977.

Fuchs, Henry. "The Automatic Sensing of 3-Dimensional Surface Points from Visual Scenes." Unpublished PhD dissertation, University of Utah, 1975.

Keppel, E. "Approximating Complex Surfaces by Triangulation of Contour Lines," Journal of Research and Development, IBM Vol. 19, No. 1 (January 1975), 2-11.

Newman, William M., and Robert F. Sproull. Principles of Interactive Computer Graphics. New York: McGraw-Hill, 1973.

APPENDIX A

COMPUTER PROGRAM

```
        COMMON/Q/NL(2),LPSTK(2,5),NPT,NJ,DZ,BD,TEKT,PE,
        1  EX(100,4),P1(100),TRIC,IPL(2,5,10),NIPL(5),IPLI,NPMAX,IP(4,2000)
        COMMON/NODE/P2(1100,3),P3(1100,3)
        COMMON/TEK/XBAR,YBAR,SF,P(2000,3),IYD2
        INTEGER LPP(40),P1,FIRST,NPI(5),NPIC(5),NPLA(2,5)
        INTEGER LT(40),C(8,8),NPL(40),NPLM(40),NCONC(40),SURR(8,40)
        REAL E(4),ID(8)
        LOGICAL FLAG(40),CLOSE,DATAF,TEKT,BD,CW,PE
        INTEGER ZS,BP,SP,EP,DP,CN,BC,TO,BO,BRANCH(10),STACK(10)
        DATA SAS,SMINT,SMAXT,WANG,DZ,SCALE/.3,.01,1.,.7,.45,.0001/
        DATA ZS,KP,L,DP,LIC,PNI,NP1,IZP/1,0,1,2,1,0,1,0/
        DATA NLA,NLP,NLV,NP,LT(1),P1(1),IYD2/0,0,0,0,0,1,0/
        DATA LPP(1),IPLI,NPMAX/1,1,2/
        DATA TEKT,CLOSE,DATAF,BD,CW/.TRUE.,.TRUE.,.TRUE.,.TRUE.,.TRUE./
        DATA NIPL/1,0,0,0,0/
        IPL(1,1,1)=1
        SMIN=(DZ*SMINT)**2.
        SMAX=(DZ*SMAXT)**2.
C   INPUT SPECS
        TYPE 1
1       FORMAT(' BRAIN DATA? ',$)
        ACCEPT 10,ANS
        IF(ANS.EQ.'Y')GO TO 2
        BD=.FALSE.
        SAS=0.
        SMIN=0.
        SMAX=10
        SCALE=1.
2       TYPE 3
3       FORMAT(' FILENAME OF INPUT DATA?  ',$)
        ACCEPT 4,INAME
4       FORMAT(A5)
        OPEN(UNIT=21,FILE=INAME)
C   INITIALIZE DISK
        READ(21,5)A,B,TZ1
        BACKSPACE 21
5       FORMAT(8G)
C   ACCEPT READ COMMANDS
        GO TO 8
6       TYPE 7,NLA
7       FORMAT(' LAST LEVEL WAS ',I2)
8       TYPE 9
9       FORMAT(' READ> ',$)
        ACCEPT 10,RC
10      FORMAT(A1)
        IF(RC.EQ.'B')GO TO 76
        IF(RC.EQ.'P')GO TO 20
        IF(RC.EQ.'L'.OR.RC.EQ.'M')GO TO 25
        IF(RC.EQ.'S')GO TO 18
        IF(RC.EQ.'T')TYPE 24,NJ,NPT
        IF(RC.EQ.'E')GO TO 139
        IF(RC.EQ.'K')GO TO 16
        IF(RC.EQ.'C')GO TO 14
        IF(RC.EQ.'D')GO TO 12
        IF(RC.EQ.'T')GO TO 8
        TYPE 11
11      FORMAT(' PARAMETERS,TOTALS,LEVEL,SCALE,EXIT,BRANCH,
        1 MANUAL,CLOSE,DEVICE,KLOCKWISE')
        GO TO 8
C   SET DEVICE
```

```
12        TYPE 13
13        FORMAT(' TEKTRONIX SCOPE? ',$)
          ACCEPT 10,ANS
          TEKT=.FALSE.
          IF(ANS.EQ.'Y')TEKT=.TRUE.
          GO TO 8
C   SET 'CLOSE' FLAG
14        TYPE 15
15        FORMAT(' CLOSE ALL LOOPS? ',$)
          ACCEPT 10,ANS
          CLOSE=.FALSE.
          IF(ANS.EQ.'Y')CLOSE=.TRUE.
C   SET CLOCKWISE FLAG
          GO TO 8
16        TYPE 17
17        FORMAT(' CLOCKWISE ORDERING? ',$)
          ACCEPT 10,ANS
          CW=.FALSE.
          IF(ANS.EQ.'Y')CW=.TRUE.
          GO TO 16
C   SET SCALE FACTOR
18        TYPE 19
19        FORMAT(' SCALE FACTOR= ',$)
          ACCEPT *,SCALE
          DZ=ZS*450.*SCALE
          SMIN=(DZ*SMINT)**2.
          SMAX=(DZ*SMAXT)**2.
          GO TO 8
C   SET PARAMETERS
20        TYPE 21
21        FORMAT(' MINIMUM SEGMENT ANGLE=  ',$)
          ACCEPT *,SANG
          SAS=SIND(SANG)
          TYPE 22
22        FORMAT(' MIN. & MAX. SEGMENT LENGTHS: ',$)
          ACCEPT *,SMINT,SMAXT
          SMIN=(DZ*SMINT)**2.
          SMAX=(DZ*SMAXT)**2.
          GO TO 8
C   TOTALS
24        FORMAT(1H ,I4,' NODES ',I4,' ELEMENTS')
C   SET SPACING AND RANGE
25        TYPE 26
26        FORMAT(' Z-SPACING=  ',$)
          READ(5,*,END=8,ERR=8)ZS
          DZ=.45*ZS
          IZP=ZS-1
          TYPE 27
27        FORMAT(' LEVEL RANGE= ',$)
          READ(5,*,END=8,ERR=8)NLS,NLF
          LIND=NLV+1
C   INITIALIZE
28        READ(21,5,END=69)(ID(J),J=1,DP),TZ
          BACKSPACE 21
          IF(DP.EQ.8)BACKSPACE 21
          IF(TZ1.EQ.TZ)GO TO 29
          NLA=NLA+1
          IF(NLA.GE.NLF)GO TO 31
          TZ1=TZ
          IZP=IZP+1
```

```
        IF((IZP.GE.ZS).AND.(NLA.GE.NLS))GO TO 31
        NP=0
        KP=0
29      KP=KP+1
C   READ FROM DISK
        READ(21,5,END=69)(ID(J),J=1,DP),Z,A,DUM,NPL(KP)
        1 ,(((P2(J+NP,I)),I=1,2),J=1,NPL(KP))
        NP=NP+NPL(KP)
        LT(KP+1)=LT(KP)+NPL(KP)
        BACKSPACE 21
C   SET DATA FILE POINTER
        DP=2*NPL(KP)+DP+12
30      DP=DP-8
        IF(DP.GT.8)GO TO 30
        GO TO 28
C   CHECK SEGMENT INTERCONNECTIVITY
31      LC=1
        NP3=0
        IF(RC.NE.'M')GO TO 37
C   GRAPHICS FOR MANUAL RE-CONSTRUCTION OF LOOPS
C   FIND WINDOW
        E(1)=P2(1,1)
        E(2)=E(1)
        E(3)=P2(1,2)
        E(4)=E(3)
        DO 32 I=2,NP
        IF(P2(I,1).LT.E(1))E(1)=P2(I,1)
        IF(P2(I,1).GT.E(2))E(2)=P2(I,1)
        IF(P2(I,2).LT.E(3))E(3)=P2(I,2)
32      IF(P2(I,2).GT.E(4))E(4)=P2(I,2)
        DX=E(2)-E(1)
        DY=E(4)-E(3)
        XBAR=(E(2)+E(1))/2.
        YBAR=(E(4)+E(3))/2.
        SF=DX
        IF(DY.GT.SF)SF=DY
        SF=700./SF
C   PAINT & LABEL SEGMENTS
        IF(.NOT.TEKT)GO TO 37
        CALL BOX
        DO 36 I=1,KP
        IND1=LT(I)+1
        IND2=LT(I+1)
        IX1=SF*(P2(IND1,1)-XBAR)+634
        IY1=SF*(P2(IND1,2)-YBAR)+390
        CALL MVTO(IX1,IY1)
        DO 33 J=IND1,IND2
        IX=SF*(P2(J,1)-XBAR)+634
        IY=SF*(P2(J,2)-YBAR)+390
33      CALL VCTO(IX,IY)
        IX=SF*(P2(IND1+1,1)-XBAR)+624
        IY=SF*(P2(IND1+1,2)-YBAR)+380
        CALL MVTO(IX,IY)
        CALL MVTO(IX,IY)
        CALL ALMODE
        TYPE 34,I
34      FORMAT(1H+,$,I2,'1')
        IX=SF*(P2(IND2-1,1)-XBAR)+624
        IY=SF*(P2(IND2-1,2)-YBAR)+380
        CALL MVTO(IX,IY)
```

```
      CALL ALMODE
      TYPE 35,I
35    FORMAT(1H+,S,I2,'2')
36    CONTINUE
      CALL MVTO(0,767)
      CALL ALMODE
37    IF(KP.EQ.1.AND.RC.NE.'M')GO TO 51
      IF(.NOT.RD)GO TO 51
      DO 38 I=1,KP
38    FLAG(I)=.FALSE.
      DO 50 IL=1,KP
      IF(FLAG(IL))GO TO 50
      JPC=1
      JLC=IL
      DO 39 I=1,NPL(IL)
      NP3=NP3+1
      DO 39 J=1,2
39    P3(NP3,J)=P2(LT(IL)+I,J)*SCALE
      NPL(LC)=NPL(IL)
C   FIND THE CLOSEST ENDPOINT
40    IF(RC.NE.'M')GO TO 43
      JP=2
      IF(JPC.EQ.2)JP=1
      TYPE 41,JLC,JP
41    FORMAT(' JOIN ',2I1,' TO ',$)
      ACCEPT 42,JLC,JPC
42    FORMAT(2I1)
      IF(JLC.NE.0)GO TO 45
      GO TO 50
43    DM=1E+35
      X1=P3(NP3,1)/SCALE
      Y1=P3(NP3,2)/SCALE
      DO 44 JL=1,KP
      IF(FLAG(JL))GO TO 44
      DO 44 JP=1,2
      IF((JL.EQ.IL).AND.(JP.EQ.2))GO TO 44
      JPR=JP
      IF(JP.EQ.2)JPR=NPL(JL)
      DIST=(X1-P2(LT(JL)+JPR,1))**2.+(Y1-P2(LT(JL)+JPR,2))**2.
      IF(DIST.GT.DM)GO TO 44
      JLC=JL
      JPC=JP
      DM=DIST
44    CONTINUE
45    FLAG(JLC)=.TRUE.
      IF(JLC.NE.IL)GO TO 46
      LC=LC+1
      GO TO 50
46    IF(JPC.EQ.2)GO TO 48
      DO 47 I=1,NPL(JLC)
      NP3=NP3+1
      DO 47 J=1,2
47    P3(NP3,J)=P2(LT(JLC)+I,J)*SCALE
      NPL(LC)=NPL(LC)+NPL(JLC)
      GO TO 40
48    DO 49 I=1,NPL(JLC)
      II=NPL(JLC)+1-I
      DO 49 J=1,2
49    P3(NP3+II,J)=P2(LT(JLC)+I,J)*SCALE
      NP3=NP3+NPL(JLC)
```

```
        NPL(LC)=NPL(LC)+NPL(JLC)
        GO TO 40
50      CONTINUE
        GO TO 53
51      DO 52 I=1,NP
        DO 52 J=1,2
52      P3(I,J)=P2(I,J)*SCALE
        LC=KP+1
C  RESET FLAGS
53      LC=LC-1
        NLV=NLV+1
        LPP(NLV+1)=LC+LPP(NLV)
C  THINNING OUT
        FIRST=0
        RZ=-(Z+51)*450*SCALE
        IF(.NOT.BD)RZ=Z*SCALE
        DO 67 M=1,LC
        RP=1
        NLP=NLP+1
        SP=2
        EP=1
        DC=0.
        N=P1(NLP)-1
        P(N+1,1)=P3(FIRST+1,1)
        P(N+1,2)=P3(FIRST+1,2)
        P(N+1,3)=RZ
C  INITIALIZE WINDOW MATRIX
        EX(NLP,1)=P3(FIRST+1,1)
        EX(NLP,2)=P3(FIRST+1,1)
        EX(NLP,3)=P3(FIRST+1,2)
        EX(NLP,4)=P3(FIRST+1,2)
        IF(NPL(M).GT.2)GO TO 54
        P1(NLP+1)=P1(NLP)+NPL(M)
        IF(NPL(M).EQ.1)GO TO 67
        P(N+2,1)=P3(FIRST+2,1)
        P(N+2,2)=P3(FIRST+2,2)
        P(N+2,3)=RZ
54      XA=P3(FIRST+RP+EP,1)-P3(FIRST+RP,1)
        YA=P3(FIRST+RP+EP,2)-P3(FIRST+RP,2)
        AL=XA*XA+YA*YA
        IF(AL.LT.SMIN)GO TO 56
55      XB=P3(FIRST+RP+EP+1,1)-P3(FIRST+RP+EP,1)
        YB=P3(FIRST+RP+EP+1,2)-P3(FIRST+RP+EP,2)
        BL=XB*XB+YB*YB
        IF(BL*AL.EQ.0.)GO TO 56
        ST=(XA*YB-XB*YA)/SQRT(AL*BL)
        IF(ST.LT.-1.)ST=-1.
        IF(ST.GT.1.)ST=1.
        IF(ABS(ST).GT.SAS)GO TO 57
C  ANGLE OR SEGMENT LENGTH  IS TOO SMALL. ELIMINATE THE NODE.
        IF(((XA+XB)**2.+(YA+YB)**2.).GT.SMAX)GO TO 57
56      EP=EP+1
        IF((RP+EP).EQ.NPL(M))GO TO 58
        GO TO 54
C  ACCEPT THE NODE.
57      P(N+SP,1)=P3(FIRST+RP+EP,1)
        P(N+SP,2)=P3(FIRST+RP+EP,2)
        P(N+SP,3)=RZ
C  SUM ANGLES TO DETERMINE DIRECTION OF ROTATION
        DC=DC+ASIN(ST)
```

```
         IF((XA*XB+YA*YB).GE.0.)GO TO 571
         IF(ST.LT.0.)DC=DC-1.5708
         IF(ST.GT.0.)DC=DC+1.5708
C  DETERMINE LOOP EXTREMES.
571      IF(P(N+SP,1).LT.EX(NLP,1))EX(NLP,1)=P(N+SP,1)
         IF(P(N+SP,1).GT.EX(NLP,2))EX(NLP,2)=P(N+SP,1)
         IF(P(N+SP,2).LT.EX(NLP,3))EX(NLP,3)=P(N+SP,2)
         IF(P(N+SP,2).GT.EX(NLP,4))EX(NLP,4)=P(N+SP,2)
         XA=XB
         YA=YB
         AL=BL
         RP=RP+EP
         SP=SP+1
         IF(RP.EQ.(NPL(M)-1))GO TO 58
         EP=1
         GO TO 55
C  ENDPOINTS
58       NSP=SP
         DO 59 I=1,2
59       IF(P3(FIRST+NPL(M),I).NE.P(N+1,I))GO TO 60
         NSP=NSP-1
         GO TO 62
60       DO 61 I=1,2
61       P(N+SP,I)=P3(FIRST+NPL(M),I)
         P(N+SP,3)=RZ
62       IF(.NOT.CLOSE)GO TO 64
         NSP=NSP+1
         DO 63 I=1,2
63       P(N+NSP,I)=P(N+1,I)
         P(N+NSP,3)=RZ
64       P1(NLP+1)=P1(NLP)+NSP
C  INSURE CLOCKWISE ORDERING
         IF(DC.LT.0..AND.CW)GO TO 67
         IF((DC.GT.0.).AND.(.NOT.CW))GO TO 67
         DO 65 I=1,NSP
         DO 65 J=1,2
65       P2(I,J)=P(N+I,J)
         DO 66 I=1,NSP
         DO 66 J=1,2
         II=NSP+1-I
66       P(N+II,J)=P2(I,J)
67       FIRST=FIRST+NPL(M)
C  RESET POINTERS
68       KP=0
         IZP=0
         NP=0
         IF(NLA.LT.NLF)GO TO 28
         GO TO 71
69       TYPE 70,NLA
70       FORMAT(' DATA ENDED AFTER LEVEL ',I2)
         DATAF=.FALSE.
71       NJ=N+NSP
C  DETERMINE CONCENTRICITY
         DO 75 M=1,NLV
         NCONC(M)=0
         LI1=LPP(M)
         LI2=LPP(M+1)-1
         DO 75 I=LI1,LI2
         DO 75 J=LI1,LI2
         IF(I.EQ.J)GO TO 75
```

```
C   CHECK FOR TRIVIAL REJECTION
          DO 72 K=1,3,2
          IF(EX(J,K+1).LT.EX(I,K))GO TO 75
72        IF(EX(J,K).GT.EX(I,K+1))GO TO 75
          I1=-1
73        I1=I1+1
C   SEE IF J SCRIBES A 360 DEGREE ARC AROUND I
          ANG=0.
          XC=P(P1(I)+I1,1)
          YC=P(P1(I)+I1,2)
          XB=P(P1(J),1)-XC
          YB=P(P1(J),2)-YC
          BL=XB*XB+YB*YB
          IF(BL.EQ.0.)GO TO 73
          DO 74 J1=P1(J),P1(J+1)-1
          XA=XB
          YA=YB
          AL=BL
          XB=P(J+1,1)-XC
          YB=P(J+1,2)-YC
          BL=XB*XB+YB*YB
          IF(BL.EQ.0.)GO TO 73
          SINE=(XA*YB-XB*YA)/SQRT(AL*BL)
74        ANG=ANG+ASIN(SINE)
          ANG=ABS(ANG)
          IF(ANG.LT.1)GO TO 75
          NCONC(M)=NCONC(M)+1
          SURR(NCONC(M),M)=I
75        CONTINUE
C   ACCEPT COMMANDS
76        TYPE 77
77        FORMAT(' BRANCH> ',$)
          ACCEPT 10,TRIC
          TRIC1=TRIC
          IF(TRIC.EQ.'A')GO TO 92
          IF(TRIC.EQ.'W')GO TO 79
          IF(TRIC.EQ.'M')GO TO 92
          IF(TRIC.EQ.'T')TYPE 24,NJ,NPT
          IF(TRIC.EQ.'I')GO TO 92
          IF(TRIC.EQ.'C')GO TO 81
          IF(TRIC.EQ.'S')GO TO 110
          IF(TRIC.EQ.'E')GO TO 139
          IF(TRIC.EQ.'T')GO TO 76
          TYPE 78
78        FORMAT(' AUTOMATIC,WARP,MANUAL,INSPECT,SINGLE,CAP,EXIT,TOTALS ')
          GO TO 76
C   CHANGE WARP ANGLE
79        TYPE 80
80        FORMAT(' MAX.WARP ANGLE= ',$)
          READ(5,*,END=76,ERR=76)ANG
          WANG=COSD(ANG)
          GO TO 76
C   CAP
81        TYPE 82
82        FORMAT(' GLOBAL LOOP NUMBER:',$)
          READ(5,114,END=76,ERR=76)N
          DO 83 I=1,NLV
          J=I
          IF(LPP(I).GE.N)GO TO 84
83        CONTINUE
```

```
84      IF(J.LT.2)GO TO 86
        TYPE 85,P(P1(LPP(J-1)),3)
85      FORMAT(' Z FOR (LEVEL-1)=',E10.3)
86      TYPE 87,P(P1(N),3)
87      FORMAT(' Z FOR (LEVEL)=',E10.3)
        IF(J.GE.NLV)GO TO 89
        TYPE 88,P(P1(LPP(J+1)),3)
88      FORMAT(' Z FOR (LEVEL+1)=',E10.3)
89      TYPE 90
90      FORMAT(' ENTER Z FOR VERTEX ',$)
        READ(5,*,END=76,ERR=76)ZV
        NJ=NJ+1
        P(NJ,1)=(EX(N,2)+EX(N,1))/2.
        P(NJ,2)=(EX(N,4)+EX(N,3))/2.
        P(NJ,3)=ZV
        IN=P1(N+1)-P1(N)-1
        B0=P1(N)-1
        DO 91 I=1,IN
        IP(1,NPT)=NJ
        IP(2,NPT)=B0+I
        IP(3,NPT)=B0+I+1
        IP(4,NPT)=0
        NPT=NPT+1
91      CONTINUE
        GO TO 76
92      TYPE 93
93      FORMAT(' START WITH WHICH LEVEL? ',$)
        READ(5,114,END=76,ERR=76)LIN
        IF(LIN.EQ.0)LIN=LIND
94      IND2=NLV-1
        IF(TRIC.EQ.'I')IND2=NLV
        IF(TRIC.NE.'A')GO TO 945
        TYPE 941
941     FORMAT(' POST-EDIT? ',$)
        ACCEPT 10,ANS
        PE=.FALSE.
        IF(ANS.EQ.'Y')PE=.TRUE.
945     DO 138 IL=LIN,IND2
        T0=LPP(IL+1)-1
        B0=LPP(IL)-1
        IF(TRIC.NE.'A')GO TO 98
C  EMPLOY CONNECTIVITY ALGORITHM
        DO 95 J=1,8
        FLAG(J)=.FALSE.
95      C(J,1)=0
        NL(1)=1
        NL(2)=1
        NLB=T0-B0
        NLT=LPP(IL+2)-LPP(IL+1)
        DO 97 J=1,NLB
        DO 97 K=1,NLT
        DO 96 L=1,3,2
        IF(EX(B0+J,L+1).LT.EX(T0+K,L))GO TO 97
96      IF(EX(B0+J,L).GT.EX(T0+K,L+1))GO TO 97
        C(J,1)=C(J,1)+1
        C(K+NLB,1)=C(K+NLB,1)+1
        C(J,C(J,1)+1)=K+NLB
        C(K+NLB,C(K+NLB,1)+1)=J
97      CONTINUE
        IF(TRIC.EQ.'A'.AND..NOT.PE)GO TO 121
```

```
C   PAINT & LABEL LOOPS
98         IF(.NOT.TEKT)GO TO 109
           DO 99 I=1,4
99         E(I)=EX(B0+1,I)
           DO 100 I=B0+2,LPP(IL+2)-1
           DO 100 J=1,3,2
           IF(E(J).GT.EX(I,J))E(J)=EX(I,J)
100        IF(E(J+1).LT.EX(I,J+1))E(J+1)=EX(I,J+1)
           DX=E(2)-E(1)
           DY=E(4)-E(3)
           XBAR=(E(2)+E(1))/2.
           YBAR=(E(4)+E(3))/2.
           SF=DX
           IF(DY.GT.SF)SF=DY
           SF=700./SF
           IND=2
           IF(TRIC.EQ.'A')GO TO 121
           CALL BOX
           IF(TRIC.EQ.'I')IND=1
101        DO 107 J=1,IND
           IND1=LPP(IL+J-1)
           IND2=LPP(IL+J)-1
           DO 107 I=IND1,IND2
           CALL MOVE(P1(I),0,0)
           DO 102 K=P1(I)+1,P1(I+1)-1
           KK=K
102        CALL DRAW(KK,0,0)
           II=I-IND1+1
           CALL MOVE(P1(I),-57,-10)
           CALL ALMODE
           IF(J.EQ.2)GO TO 105
           IF(TRIC.EQ.'I')TYPE 103,I
103        FORMAT(1H+,S,I3)
           IF(TRIC.NE.'I') TYPE 104,II
104        FORMAT(1H+,S,I3,'B')
           GO TO 107
105        TYPE 106,II
106        FORMAT(1H+,$,I3,'T')
107        CONTINUE
           CALL MVTO(0,767)
           CALL ALMODE
           TYPE 108,IL
108        FORMAT(' >> LEVEL ',I2)
           IF(TRIC.NE.'I')GO TO 109
           READ(5,114,END=76,ERR=76)ICONT
           GO TO 138
109        NL(1)=0
           NL(2)=0
C   SELECT CONNECTIVITY MANUALLY
110        TYPE 111
111        FORMAT(' BOTTOM LOOP(S) (LOCAL):')
           GO TO 113
           TYPE 112
112        FORMAT(' BOTTOM LOOPS:(GLOBAL)')
113        READ(5,114,END=76,ERR=76),N
114        FORMAT(I)
           IF(N.EQ.0)GO TO 115
           NL(1)=NL(1)+1
           LPSTK(1,NL(1))=N+B0
           IF(TRIC.EQ.'S')LPSTK(1,NL(1))=N
```

```
      GO TO 113
115   IF(NL(1).EQ.0)GO TO 119
      TYPE 116
116   FORMAT(' TOP LOOP(S)')
117   READ(5,114,END=76,ERR=76),N
      IF(N.EQ.0)GO TO 118
      NL(2)=NL(2)+1
      LPSTK(2,NL(2))=N+T0
      IF(TRIC.EQ.'S')LPSTK(2,NL(2))=N
      GO TO 117
118   IF(NL(2).EQ.0)GO TO 119
      CALL QUAD
      TRIC=TRIC1
      IF(TRIC.EQ.'A')GO TO 138
119   TYPE 120
120   FORMAT(' NEXT LEVEL?',$)
      ACCEPT 10,ANS
      IF(ANS.EQ.'Y')GO TO 138
      IF(.NOT.TEKT)GO TO 109
      CALL BOX
      GO TO 101
C  AUTOMATIC BRANCHING ALGORITHM
121   NL(1)=1
      NL(2)=1
      IF(NCONC(IL).EQ.0)GO TO 128
      DO 127 J=1,NCONC(IL)
      K=SURR(J,IL)-B0
      FLAG(K)=.TRUE.
      IF(NCONC(IL+1).GT.0)GO TO 123
      IF(C(K,1).GT.1)TYPE 122,IL
122   FORMAT(' ?ILLEAGLE CONCENTRICITY IN LEVEL',I2)
      LPSTK(1,1)=K+B0
      LPSTK(2,1)=C(K,2)+T0
      CALL QUAD
      GO TO 127
123   DO 124 L=1,C(K,1)
      DO 124 N=1,NCONC(IL+1)
      IF(SURR(N,IL+1).EQ.C(K,L+1))GO TO 125
124   CONTINUE
      LPSTK(2,1)=C(K,2)+T0
      GO TO 126
125   LPSTK(2,1)=SURR(N,IL+1)
      FLAG(SURR(N,IL+1)-T0)=.TRUE.
126   CALL QUAD
127   CONTINUE
128   IF(NCONC(IL+1).EQ.0)GO TO 130
      DO 129 J=1,NCONC(IL+1)
      K=SURR(J,IL)-B0
      FLAG(K)=.TRUE.
      IF(C(K,1).GT.1)TYPE 122,IL
      LPSTK(1,1)=K+B0
      LPSTK(2,1)=C(K,2)+B0
129   CALL QUAD
C  PREPARE REMAINING LOOPS FOR QUAD
130   DO 138 J=1,NLB
      IF(FLAG(J))GO TO 138
      FLAG(J)=.TRUE.
      IF(C(J,1)-1)138,131,132
C  SIMPLE 1 ON 1
131   IV=C(C(J,1)+NLB,1)
```

```
      IF(IV.GT.1)GO TO 132
      FLAG(C(J,2))=.TRUE.
      NL(1)=1
      NL(2)=1
      LPSTK(2,1)=C(J,2)+T0-NLB
      LPSTK(1,1)=C(C(J,2),2)+B0
      CALL QUAD
      GO TO 138
C  DETERMINE EXTENT OF BRANCHING
132   SP=1
      NBR=1
      BRANCH(1)=J
      STACK(1)=J
133   DO 134 I=1,C(STACK(SP),1)
      NODE=C(STACK(SP),I+1)
      IF(FLAG(NODE))GO TO 134
      FLAG(NODE)=.TRUE.
      NBR=NBR+1
      BRANCH(NBR)=NODE
      IF(C(NODE,1).EQ.1)GO TO 134
      SP=SP+1
      STACK(SP)=NODE
      GO TO 133
134   CONTINUE
      IF(SP.EQ.1)GO TO 135
      SP=SP-1
      GO TO 133
135   NL(1)=0
      NL(2)=0
      DO 137 I=1,NBR
      IF(BRANCH(I).LE.NLB)GO TO 136
      NL(2)=NL(2)+1
      LPSTK(2,NL(2))=BRANCH(I)+T0-NLB
      GO TO 137
136   NL(1)=NL(1)+1
      LPSTK(1,NL(1))=BRANCH(I)+B0
137   CONTINUE
      CALL QUAD
138   CONTINUE
      IF(TRIC.EQ.'I')GO TO 76
      LIND=NLV+1
      IF(DATAF)GO TO 6
C  OUTPUT
139   TYPE 140
140   FORMAT(' DONE? ',$)
      ACCEPT 10,ANS
      IF(ANS.NE.'Y')GO TO 8
      NPT=NPT-1
      NP=NPMAX-1
      TYPE 24,NJ,NPT
      IPL(2,IPLI,NIPL(IPLI))=NPT
      TYPE 141
141   FORMAT(' OUTPUT FILENAME?  ',$)
      ACCEPT 4,ONAME
      OPEN(UNIT=22,FILE=ONAME)
      DO 142 I=1,NP
      NPIC(I)=0
      DO 142 J=1,NIPL(I)
142   NPIC(I)=NPIC(I)+IPL(2,I,J)-IPL(1,I,J)+1
      WRITE(22,146)NP,NJ,NPT
```

```
      N=0
      DO 143 I=1,NP
      J=N+1
      N=J+NPIC(I)-1
      NPLA(1,I)=J
143   NPLA(2,I)=N
      WRITE(22,146)((NPLA(I,J),I=1,2),J=1,NP)
      WRITE(22,147)((P(I,J),J=1,3),I=1,NJ)
      WRITE(22,146)((((IP(I,J),I=1,4),J=IPL(1,N,M),IPL(2,N,M))
      1 ,M=1,NIPL(N)),N=1,NP)
      TYPE 144
144   FORMAT(' SPECIAL FUNCTION FILE?',$)
      ACCEPT 10,ANS
      IF(ANS.NE.'Y')GO TO 148
      TYPE 145
145   FORMAT(' FILENAME OF S.F. FILE: ',$)
      ACCEPT 4,SNAME
      OPEN(UNIT=23,FILE=SNAME)
      WRITE(23,147)(P(I,3),I=1,NJ)
146   FORMAT(20I4)
147   FORMAT(6E12.5)
148   STOP
      END
      SUBROUTINE QUAD
      LOGICAL FLAG1,FLAG2,FLAG(5),CLOSE,IPF(10),DRAWF,MAP,BD,TEKT,PE,PEN
      COMMON/Q/NL(2),LPSTK(2,5),NPT,NJ,DZ,BD,TEKT,PE,
      1  EX(100,4),P1(100),TRIC,IPL(2,5,10),NIPL(5),IPLI,NPMAX,IP(4,2000)
      COMMON/NODE/P2(2,1100,3)
      COMMON/TEK/XBAR,YBAR,SF,P(2000,3),IYD2
      INTEGER TEMP(200),NV(2),NEW(2,5),V(2,200),FIRST,P1,PDUB(12)
      REAL TRANS(2,4),TMAP(2,2),S(2)
      INTEGER ORDER(5,5),CP(5,5),O(5),FIRST1,FIRST2,CL,CT
      REAL D(5,5),DT(5),E(4)
      INTEGER NCP(2),CV(2,150),N1(2),N2(2)
      DATA IVIEW,MAP,IND6/1,.TRUE.,2/
      NPT1=NPT
      NPT2=NPT
      NNEW=0
      NTEMP=0
      XBART=XBAR
      YBART=YBAR
      NTMAX=1
      NBMAX=1
      NLINE=767
      PEN=.FALSE.
      SFT=SF
      DRAWF=.FALSE.
      DO 48 M=1,2
      O(1)=1
      IF(NL(M).EQ.1)GO TO 29
      IF(TRIC.NE.'A'.OR.PE)GO TO 12
C   AUTOMATIC:
C    FIND CLOSEST DISTANCE BETWEEN ALL LOOPS
      DO 7 I=1,NL(M)-1
      FIRST1=P1(LPSTK(M,I))
      LAST1=P1(LPSTK(M,I)+1)-1
      DO 7 J=I+1,NL(M)
      FIRST2=P1(LPSTK(M,J))
      LAST2=P1(LPSTK(M,J)+1)-1
      CP(I,J)=FIRST1
```

```
        CP(J,I)=FIRST2
        DIST=1E35
        FLAG1=.FALSE.
1       FLAG2=.TRUE.
        DO 3 K=FIRST1,LAST1
        TDIST=(P(K,1)-P(CP(J,I),1))**2+(P(K,2)-P(CP(J,I),2))**2
        IF(TDIST.GE.DIST)GO TO 3
        DO 2 L=1,NL(M)
2       IF(K.EQ.CP(I,L))GO TO 3
        DIST=TDIST
        FLAG2=.FALSE.
        CP(I,J)=K
3       CONTINUE
        IF(FLAG2.AND.FLAG1)GO TO 6
        FLAG1=.TRUE.
        DO 5 K=FIRST2,LAST2
        TDIST=(P(K,1)-P(CP(I,J),1))**2+(P(K,2)-P(CP(I,J),2))**2
        IF(TDIST.GE.DIST)GO TO 5
        DO 4 L=1,NL(M)
4       IF(K.EQ.CP(J,L))GO TO 5
        DIST=TDIST
        FLAG1=.FALSE.
        CP(J,I)=K
5       CONTINUE
        IF(FLAG1.AND.FLAG2)GO TO 6
        GO TO 1
6       D(J,I)=DIST
7       D(I,J)=DIST
C  PICK THE PROPER LOOP SEQUENCE
        DO 9 I=1,NL(M)
        J1=I
        DT(I)=0.
        DO 8 J=1,5
8       FLAG(J)=.FALSE.
        DIST=0.
        DO 9 J=1,NL(M)
        FLAG(J1)=.TRUE.
        DT(I)=DT(I)+DIST
        ORDER(I,J)=J1
        J2=J1
        DIST=1E35
        DO 9 K=1,NL(M)
        IF(FLAG(K))GO TO 9
        IF(DIST.LE.D(J2,K))GO TO 9
        J1=K
        DIST=D(J2,K)
9       CONTINUE
        DIST=1E35
        DO 10 I=1,NL(M)
        IF(DIST.LE.DT(I))GO TO 10
        DIST=DT(I)
        IT=I
10      CONTINUE
        DO 11 J=1,NL(M)
11      O(J)=ORDER(IT,J)
C  DISLPAY LOOPS
12      IF(TRIC.EQ.'A'.AND..NOT.PE)GO TO 26
        IF(.NOT.TEKT)GO TO 17
        CALL BOX
        DO 16 K=1,2
```

```
        DO 16 J=1,NL(K)
        IND1=P1(LPSTK(K,J))
        IND2=P1(LPSTK(K,J)+1)-1
        CALL MOVE(IND1,0,0)
        DO 13 L=IND1+1,IND2
        LL=L
13      CALL DRAW(LL,0,0)
        IF(M.NE.K)GO TO 16
        CALL ALMODE
        TYPE 14,J
14      FORMAT(1H+,$,'LOOP',I1)
        DO 16 L=IND1+2,IND2,2
        LL=L
        CALL CROSS(LL,3)
        CALL MOVE(LL,-57,-10)
        CALL ALMODE
        TYPE 15,L
15      FORMAT(1H+,$,I4)
16      CONTINUE
        CALL MVTO(0,767)
        CALL ALMODE
17      TYPE 18
18      FORMAT(' ENTER LOOP SEGUENCE')
        N=0
        DO 19 L=1,NL(M)
        N=N+1
19      READ(5,*,END=137,ERR=137)O(L)
20      NL(M)=N
        IF(N.EQ.1)GO TO 26
        DO 23 L=1,NL(M)-1
        TYPE 21,O(L)
21      FORMAT(' L',I1,' P?',$)
        ACCEPT *,CP(O(L),O(L+1))
        TYPE 22,O(L+1)
22      FORMAT(1H+,$,' TO L',I1,' P?',$)
23      ACCEPT *,CP(O(L+1),O(L))
        TYPE 24
24      FORMAT(' CHANGES?',$)
        ACCEPT 25,ANS
25      FORMAT(A1)
        IF(ANS.EQ.'Y')GO TO 17
C  INTERPOLATE NEW POINTS
26      IF(NL(M).EQ.1)GO TO 29
        I=LPSTK(2,1)
261     J=LPSTK(1,1)
262     I1=P1(I)
263     J1=P1(J)
264     ZI=P(I1,3)
265     ZJ=P(J1,3)
266     Z=(ZI+ZJ)/2.
        DO 28 I=1,NL(M)-1
        NJ=NJ+1
        NEW(M,I)=NJ
        DO 27 J=1,2
        K=J
270     I2=CP(O(I),O(I+1))
271     I3=CP(O(I+1),O(I))
272     TJ2=P(I2,J)
273     TJ3=P(I3,J)
274     TJ=(TJ2+TJ3)/2.
```

```
27        P(NJ,J)=TJ
28        P(NJ,3)=Z
C   COMBINE LOOPS
29        FLAG1=.TRUE.
          FLAG2=.FALSE.
          NV(M)=0
          IND1=1
          IND2=NL(M)
          IND3=1
30        DO 43 I=IND1,IND2,IND3
31        FIRST=P1(LPSTK(M,O(I)))-1
          IF(NL(M).EQ.1)GO TO 39
          CL=CP(O(I),O(I+1))-FIRST
          IF(I.EQ.1.AND.IND3.EQ.-1)GO TO 38
          IF(I.EQ.1)GO TO 32
          CT=CP(O(I),O(I-1))-FIRST
          IF(I.EQ.IND2.AND.IND3.EQ.1)GO TO 35
          IF(.NOT.FLAG1)GO TO 33
32        FLAG1=.FALSE.
          FLAG2=.TRUE.
          N=CL
          IF(IND3)N=CT
          GO TO 40
33        IF(IND3)36,34
34        IF(CL.GT.CT)GO TO 37
          FLAG1=.TRUE.
35        FIRST=FIRST+CT-1
          N=P1(LPSTK(M,O(I))+1)-FIRST-2
          GO TO 40
36        IF(CL.GT.CT)GO TO 38
          FIRST=FIRST+CL-1
          N=CT-CL+1
          FLAG2=.TRUE.
          GO TO 40
37        FIRST=FIRST+CT-1
          FLAG2=.TRUE.
          N=CL-CT+1
          GO TO 40
38        FIRST=FIRST+CL-1
          FLAG1=.TRUE.
39        IDUM=LPSTK(M,O(I))+1
          N=P1(IDUM)-FIRST-2
C   ALL THAT WORK FOR THIS LITTLE LOOP!
40        IF(I.EQ.IND2)FLAG1=.FALSE.
          IF(N.EQ.0)N=1
          DO 41 J=1,N
          NV(M)=NV(M)+1
41        V(M,NV(M))=FIRST+J
          IF(.NOT.FLAG2)GO TO 42
          N=NEW(M,I)
          IF(IND3.EQ.-1)N=NEW(M,I-1)
          NV(M)=NV(M)+1
          V(M,NV(M))=N
          FLAG2=.FALSE.
42        IF(FLAG1)GO TO 31
43        CONTINUE
          IF(NL(M).EQ.1)GO TO 44
          IF(IND3.EQ.-1)GO TO 44
          FLAG1=.TRUE.
          IND3=-1
```

```
      IND1=NL(M)
      IND2=1
      GO TO 30
44    CONTINUE
C  SEGMENT ENDPOINTS
      NV(M)=NV(M)+1
      GO TO 46
      DO 45 I=1,2
45    IF(P(V(M,1),I).NE.P(V(M,NV(M)),I))GO TO 47
46    V(M,NV(M))=V(M,1)
      GO TO 48
47    V(M,NV(M))=V(M,NV(M)-1)+1
48    CONTINUE
      IF(TRIC.EQ.'A')TRI='A'
      IF(TRIC.EQ.'A'.AND..NOT.PE)GO TO 111
C  DETERMINE EXTREMES
      DO 49 I=1,4
49    E(I)=EX(LPSTK(1,1),I)
      DO 50 M=1,2
      DO 50 J=1,NL(M)
      DO 50 I=1,3,2
      ET1=EX(LPSTK(M,J),I)
      ET2=EX(LPSTK(M,J),I+1)
      IF(ET1.LT.E(I))E(I)=ET1
50    IF(ET2.GT.E(I+1))E(I+1)=ET2
      DX=E(2)-E(1)
      DY=E(4)-E(3)
      DYT=DY+DZ
      SF2=DX
      IF(DYT.GT.SF2)SF2=DYT
      SF2=700./SF2
      SF1=DX
      IF(DY.GT.SF1)SF1=DY
      SF1=700./SF1
      XBAR=(E(1)+E(2))/2.
      YBAR=(E(3)+E(4))/2.
51    IF(IVIEW.EQ.1)GO TO 52
      SF=SF2
      IYD2=390.*DZ/DYT
      GO TO 53
52    SF=SF1
      IYD2=0
53    IF(.NOT.TEKT)GO TO 63
      CALL BOX
      DO 54 M=1,2
      CALL CROSS(V(M,1),5)
      DO 54 J=2,NV(M)
54    CALL DRAW(V(M,J),0,0)
      IF(IVIEW.EQ.1)GO TO 57
      IY=10
      DO 56 M=1,2
      IX=SF*(P(V(M,1),1)-XBAR)+634
      CALL MVTO(IX,IY)
      DO 55 J=2,NV(M)
      IX=SF*(P(V(M,J),1)-XBAR)+634
55    CALL VCTO(IX,IY)
56    IY=SF*DX+10
57    IF(NPT1.EQ.NPT)GO TO 60
      IND=NPT1+1
      DO 58 I=IND,NPT
```

```
        CALL MOVE(IP(1,I),0,0)
58      CALL DRAW(IP(2,I),0,0)
        IF(IVIEW.EQ.1)GO TO 60
        IYT=4*IYD2/39
        IYB=35*IYD2/39
        DO 59 I=IND,NPT
        IX=SF*(P(IP(1,I),1)-XBAR)+634
        CALL MVTO(IX,IYB)
        IX=SF*(P(IP(2,I),1)-XBAR)+634
59      CALL VCTO(IX,IYT)
60      IF(TRIC.EQ.'A'.AND..NOT.PEN)GO TO 111
        CALL MOVE(V(1,1),-10,-10)
        CALL ALMODE
        TYPE 600
600     FORMAT(1H+,$,'B')
        CALL MOVE(V(2,1),-10,-10)
        TYPE 601
601     FORMAT(1H+,$,'T')
        DO 62 M=1,2
        J=NV(M)-2
        DO 62 I=M,J,IND6
        CALL MOVE(V(M,I),-45,-10)
        CALL ALMODE
        TYPE 61,I
61      FORMAT(1H+,$,I3)
62      CONTINUE
        CALL MVTO(0,767)
        NLINE=767
        CALL ALMODE
63      IF(DRAWF)GO TO 94
C  ACCEPT COMMANDS
64      TYPE 65
        NLINE=NLINE-20
65      FORMAT(' TRIANGULATE>',$)
        ACCEPT 136,TRI
        IF(TRI.EQ.'A')GO TO 111
        IF(TRI.EQ.'V')GO TO 72
        IF(TRI.EQ.'R')GO TO 90
        IF(TRI.EQ.'C')GO TO 137
        IF(TRI.EQ.'T')GO TO 73
        IF(TRI.EQ.'I')GO TO 94
        IF(TRI.EQ.'D')GO TO 69
        IF(TRI.EQ.'P')GO TO 75
        IF(TRI.EQ.'E')GO TO 71
        IF(TRI.EQ.'O')GO TO 83
        IF(TRI.EQ.'M')GO TO 67
        IF(TRI.EQ.'U')GO TO 68
        TYPE 66
        NLINE=NLINE-80
66      FORMAT(' AUTO,INTERACTIVE,',/,' RENUMBER,DENSITY,',/,' CONT,
        1 TOTALS',/,' PART,ERASE,VIEW,MAP,UNMAP,ONE')
        GO TO 64
C  SET MAP FLAG
67      MAP=.TRUE.
        GO TO 64
C  RELEASE MAP FLAG
68      MAP=.FALSE.
        GO TO 64
C  SET NUMBER DENSITY
69      TYPE 70
```

```
70          FORMAT(' IND3=',$)
            ACCEPT *,IND6
            GO TO 53
C    ERASE
71          NPT=NPT1
            NTMAX=1
            NBMAX=1
            DRAWF=.FALSE.
            GO TO 53
C    CHANGE NUMBER OF ORTHOGRAPHIC VIEWS
72          I=IVIEW
            IF(I.EQ.2)IVIEW=1
            IF(I.EQ.1)IVIEW=2
            GO TO 51
C    TOTALS
73          TYPE 74,NJ,NPT
74          FORMAT(' ',I4,' NODES, ',I4,' ELEMENTS')
            GO TO 64
75          TYPE 76,IPLI
C    CHANGE PART NUMBER
76          FORMAT(' CHANGE PART FROM ',I1,' TO ',$)
            ACCEPT *,N
            IF(N.LE.NPMAX.AND.NPT.GT.IPL(1,IPLI,NIPL(IPLI)))GO TO 80
            IF(N.GT.NPMAX)GO TO 78
            TYPE 77
77          FORMAT(' CANT CHANGE PART NUMBER YET')
            GO TO 64
78          TYPE 79,NPMAX
79          FORMAT(' MUST BE LESS THAN ',I2)
            GO TO 75
80          IF(IPLI.EQ.N)GO TO 64
            I=NPT-1
            IF(IPL(1,IPLI,NIPL(IPLI)).LT.I)GO TO 81
            NIPL(IPLI)=NIPL(IPLI)-1
            GO TO 82
81          IPL(2,IPLI,NIPL(IPLI))=I
82          NIPL(N)=NIPL(N)+1
            IPL(1,N,NIPL(N))=NPT
            IF(N.EQ.NPMAX)NPMAX=NPMAX+1
            IPLI=N
            GO TO 64
C    ONE AT A TIME
83          DO 88 I=1,4
            NLINE=NLINE-20
            TYPE 84,I
84          FORMAT(' POINT ',I1,'. T/B,NO.',$)
            ACCEPT 85,T,N
85          FORMAT(A1,I)
            IF(T.EQ.'T')GO TO 86
            IF(T.EQ.'B')GO TO 87
            IF(I.NE.4)GO TO 64
            IP(4,NPT)=0
            GO TO 88
86          IP(I,NPT)=V(2,N)
            GO TO 88
87          IP(I,NPT)=V(1,N)
88          CONTINUE
            CALL MOVE(IP(1,NPT),0,0)
            CALL DRAW(IP(2,NPT),0,0)
            CALL DRAW(IP(3,NPT),0,0)
```

```
        IF(IP(4,NPT).EQ.0)CALL DRAW(IP(4,NPT),0,0)
89      NPT=NPT+1
        CALL MVTO(0,NLINE)
        CALL ALMODE
        GO TO 64
C  RENUMBER
90      TYPE 91
91      FORMAT(' NEW #1 NODE',/,' (TOP CONTOUR )',$)
        NLINE=NLINE-40
        DRAWF=.FALSE.
        ACCEPT *,N5
        IF(N5.LT.1)GO TO 90
        GO TO 113
94      NTMT=NTMAX
        NBMT=NBMAX
        TYPE 95,NTMAX,NV(2)
        NLINE=NLINE-80
95      FORMAT(' TOP:',I3,'-',I3)
        READ(5,96,END=64,ERR=64),N1(2),N2(2)
96      FORMAT(2I)
        IF(N1(2).EQ.0)GO TO 137
        TYPE 97,NBMAX,NV(1)
97      FORMAT(' BOTTOM:',I3,'-',I3)
        READ(5,96,END=64,ERR=64),N1(1),N2(1)
        NPT2=NPT
98      IF(TRI.NE.'A')GO TO 99
        N1(1)=1
        N1(2)=1
        N2(1)=NV(1)
        N2(2)=NV(2)
99      NTDEL=1
        IF(N1(2).GT.N2(2))NTDEL=-1
        NBDEL=1
        IF(N1(1).GT.N2(1))NBDEL=-1
        IF(.NOT.MAP)GO TO 108
        DO 100 I=1,2
100     IF(N1(I).EQ.N2(I))GO TO 110
        DO 102 I=1,2
        IND1=N1(I)
        IND2=N2(I)
        IF(IND1.LT.IND2)GO TO 101
        IND1=IND2
        IND2=N1(I)
101     TRANS(I,1)=P(V(I,IND1),1)
        TRANS(I,2)=TRANS(I,1)
        TRANS(I,3)=P(V(I,IND1),2)
        TRANS(I,4)=TRANS(I,3)
        DO 102 J=IND1+1,IND2
        PX=P(V(I,J),1)
        IF(PX.LT.TRANS(I,1))TRANS(I,1)=PX
        IF(PX.GT.TRANS(I,2))TRANS(I,2)=PX
        PY=P(V(I,J),2)
        IF(PY.LT.TRANS(I,3))TRANS(I,3)=PY
102     IF(PY.GT.TRANS(I,4))TRANS(I,4)=PY
103     DO 104 I=1,2
        TMAP(1,I)=(TRANS(I,2)+TRANS(I,1))/2.
104     TMAP(2,I)=(TRANS(I,3)+TRANS(I,4))/2.
        DO 105 I=1,2
        IF(TRANS(I,2).EQ.TRANS(I,1))GO TO 108
105     IF(TRANS(I,3).EQ.TRANS(I,4))GO TO 108
```

```
C  MAP
         DO 107 I=1,2
         S(1)=100./(TRANS(I,2)-TRANS(I,1))
         S(2)=100./(TRANS(I,4)-TRANS(I,3))
         IN=NBDEL
         IF(I.EQ.2)IN=NTDEL
         DO 107 J=N1(I),N2(I),IN
         DO 106 K=1,2
106      P2(I,J,K)=S(K)*(P(V(I,J),K)-TMAP(K,I))
107      P2(I,J,3)=P(V(I,J),3)
         GO TO 110
C  DON'T MAP
108      DO 109 I=1,2
         IN=NBDEL
         IF(I.EQ.2)IN=NTDEL
         DO 109 J=N1(I),N2(I),IN
         DO 109 K=1,3
109      P2(I,J,K)=P(V(I,J),K)
110      NT=N1(2)
         NB=N1(1)
         NTMAX=N2(2)
         NBMAX=N2(1)
         GO TO 117
111      D1=1E35
         X1=P(V(1,1),1)
         Y1=P(V(1,1),2)
         DO 112 I=1,NV(2)
         D2=(P(V(2,I),1)-X1)**2+(P(V(2,I),2)-Y1)**2
         IF(D2.GE.D1)GO TO 112
         D1=D2
         N5=I
112      CONTINUE
C  RE-ORDER
113      IF(N5.EQ.1)GO TO 116
         DO 114 J=1,NV(2)
114      TEMP(J)=V(2,J)
         DO 115 J=1,NV(2)
         JR=J+N5-1
         IF(JR.GT.NV(2))JR=JR-NV(2)+1
115      V(2,J)=TEMP(JR)
116      IF(TRI.EQ.'R')GO TO 53
         GO TO 98
C  TRIANGULATE BETWEEN LIMITS
117      IFLAG=0
118      IP(1,NPT)=V(2,NT)
         IP(2,NPT)=V(1,NB)
         IP(4,NPT)=0
         RX=P2(2,NT,1)-P2(1,NB,1)
         RY=P2(2,NT,2)-P2(1,NB,2)
         RZ=P2(2,NT,3)-P2(1,NB,3)
         IF(NT.EQ.NTMAX)GO TO 125
         IF(NB.EQ.NBMAX)GO TO 120
         D1=0.
         D2=0.
         DO 119 I=1,3
         D1=D1+(P2(2,NT,I)-P2(1,NB+NBDEL,I))**2
119      D2=D2+(P2(2,NT+NTDEL,I)-P2(1,NB,I))**2
C  SELECT SHORTEST DIAGONAL
         IF(D1.LT.D2)GO TO 125
C  TOP NODE IS CLOSEST
```

```
120     NT=NT+NTDEL
        IP(3,NPT)=V(2,NT)
121     NPT=NPT+1
        SX=P2(2,NT,1)-P2(1,NB,1)
        SY=P2(2,NT,2)-P2(1,NB,2)
        SZ=P2(2,NT,3)-P2(1,NB,3)
        X1=RY*SZ-RZ*SY
        Y1=RZ*SX-RX*SZ
        Z1=RX*SY-RY*SX
        AREA1=(X1*X1+Y1*Y1+Z1*Z1)
        IF(AREA1.NE.0)GO TO 122
        NPT=NPT-1
        GO TO 118
122     IF(IFLAG.EQ.2)GO TO 124
123     IFLAG=1
        GO TO 118
124     ANGT=(X1*X2+Y1*Y2+Z1*Z2)/SQRT(AREA1*AREA2)
        IF(ANGT.LT.WANG)GO TO 123
        IFLAG=0
        IP(4,NPT-2)=V(2,NT)
        NPT=NPT-1
        GO TO 118
C  BOTTOM NODE IS CLOSEST
125     IF(NB.EQ.NBMAX)GO TO 130
        NB=NB+NBDEL
        IP(3,NPT)=V(1,NB)
126     NPT=NPT+1
        SX=P2(1,NB,1)-P2(1,NB-NBDEL,1)
        SY=P2(1,NB,2)-P2(1,NB-NBDEL,2)
        SZ=P2(1,NB,3)-P2(1,NB-NBDEL,3)
        X2=RY*SZ-RZ*SY
        Y2=RZ*SX-RX*SZ
        Z2=RX*SY-RY*SX
        AREA2=(X2*X2+Y2*Y2+Z2*Z2)
        IF(AREA2.NE.0.)GO TO 127
        NPT=NPT-1
        GO TO 118
127     IF(IFLAG.EQ.1)GO TO 129
128     IFLAG=2
        GO TO 118
129     ANGT=(X1*X2+Y1*Y2+Z1*Z2)/SQRT(AREA1*AREA2)
        IF(ANGT.LT.WANG)GO TO 128
        IP(4,NPT-2)=IP(1,NPT-1)
        IP(3,NPT-2)=IP(3,NPT-1)
        NPT=NPT-1
        IFLAG=0
        GO TO 118
130     IF(TRIC.EQ.'A'.AND..NOT.PE)GO TO 137
C  PAINT NEW DIAGONALS
        IF(.NOT.TEKT)GO TO 134
131     DO 132 I=NPT2,NPT
        CALL MOVE(IP(1,I),0,0)
132     CALL DRAW(IP(2,I),0,0)
        IF(IVIEW.EQ.1)GO TO 134
        IYT=4*IYT2/39
        IYB=35*IYT2/39
        DO 133 I=NPT2,NPT
        IX=SF*(P(IP(1,I),1)-XBAR)+634
        CALL MVTO(IX,IYB)
        IX=SF*(P(IP(2,I),1)-XBAR)+634
```

```
133     CALL VCTO(IX,IYT)
134     NLINE=NLINE-100
        CALL MVTO(0,NLINE)
        CALL ALMODE
        TYPE 135
135     FORMAT(' CHANGES?',$)
        ACCEPT 136,ANS
        DRAWF=.TRUE.
        IF(NLINE.LT.200)GO TO 53
136     FORMAT(A1)
        IF(ANS.EQ.'Y')GO TO 138
        IF(TRI.EQ.'A')GO TO 137
        IF(NTMAX.EQ.NV(2).AND.NBMAX.EQ.NV(1))GO TO 137
        GO TO 94
138     NTMAX=NTMT
        NBMAX=NBMT
        NPT=NPT2
        IF(TRI.EQ.'A')DRAWF=.FALSE.
        IF(TRIC.NE.'A')GO TO 53
        PEN=.TRUE.
        TYPE 139
139     FORMAT(' CHANGE CONNECTIVITY? ',$)
        ACCEPT 136,ANS
        IF(ANS.NE.'Y')GO TO 53
        TRIC='M'
C  RESTORE PARAMETERS BEFORE RETURNING
137     YBAR=YBART
        XBAR=XBART
        IYD2=0
        SF=SFT
        RETURN
        END

        SUBROUTINE BOX
        CALL CLHOA
        CALL MVTO(244,779)
        CALL VCTO(1023,779)
        CALL VCTO(1023,0)
        CALL VCTO(244,0)
        CALL VCTO(244,779)
        RETURN
        END
        SUBROUTINE CROSS(I,ISIZE)
        COMMON/TEK/XBAR,YBAR,SF,P(2000,3),IYD2
        CALL MOVE(I,-ISIZE,0)
        CALL DRAW(I,ISIZE,0)
        CALL MOVE(I,0,ISIZE)
        CALL DRAW(I,0,-ISIZE)
        CALL MOVE(I,0,0)
        RETURN
        END
        SUBROUTINE MOVE(I,IXD,IYD)
        COMMON/TEK/XBAR,YBAR,SF,P(2000,3),IYD2
        IX=SF*(P(I,1)-XBAR)+634+IXD
        IY=SF*(P(I,2)-YBAR)+390+IYD+IYD2
        CALL MVTO(IX,IY)
        RETURN
        END
        SUBROUTINE DRAW(I,IXD,IYD)
        COMMON/TEK/XBAR,YBAR,SF,P(2000,3),IYD2
```

```
IX=SF*(P(I,1)-XBAR)+634+IXD
IY=SF*(P(I,2)-YBAR)+390+IYD+IYD2
CALL VCTO(IX,IY)
RETURN
END
```

APPENDIX B

USER DOCUMENTATION

Throughtout this documentation, cues typed by the computer are underlined. The program commences by asking:

BRAIN DATA?

User replies "yes", or "no" (brain data is handled slightly differently than user generated data). Next, the program requests

FILE NAME OF INPUT DATA?

Whereupon the user types the file name. The brain data files are called Bn.DAT where n is an integer from 1 to 22. Now the program requests a command by giving the prompt:

READ>

This prompt is so named because it is here that parameters are set preparatory to reading the data files. READ> responds to the following commands, of which only the first letter is required.

PARAMETERS -	Change node elimination parameters S_{min}, S_{max}, and θ_{min}.
TOTALS -	Causes the current number of nodes and panels to be output on the terminal.
SCALE -	Enables change of scale factor. Default is .0001 for brain data and 1 for non-brain data.
CLOSE -	Computer responds, CLOSE ALL LOOPS? If yes, every contour line will be treated as a closed loop. Otherwise, only contour lines whose first and last nodes agree will be considered closed. Default is "yes".
DEVICE -	Computer responds: TEKTRONIX SCOPE? "Yes" enables graphical output. "No" prohibits it. Default is "yes".
CLOCKWISE -	Computer responds: CLOCKWISE ORDERING? "Yes" forces all loops to run clockwise. "No" forces counterclockwise ordering. Default is "yes". Clockwise loops result in counterclockwise

definition of visible panels. This understanding is useful in using the Poor Man's hidden surface elimination in MOVIE.BYU.

EXIT - Exits from the program.

LEVEL - Since there are no convenient default values for the parameters set by the LEVEL command, this is the command which actually initiates the reading of the disk file - after the following two questions are answered: Z SPACING= . User responds with an integer "n" which indicates that every "nth" contour level is to be read in from disk. The computer then asks: LEVEL RANGE:.The user enters two integers, j and k. Thereupon, the algorithm procedes to read every nth contour level from j to k inclusive, imposing the node elimination parameters. When array P has been loaded, ready for triangulation, the program issues the BRANCH> prompt.

BRANCH - Proceeds to the BRANCH> prompt without reading from disk.

MANUAL - Same as LEVEL, except loops from the brain data can be constructed interactively from their constituent segments. This command is never needed except in cases where closed and non-closed contour lines coincide on the same level.

The commands of PARAMETERS, TOTALS, SCALE, KLOCKWISE, CLOSE, and DEVICE return to the READ> prompt. LEVEL, BRANCH and MANUAL proceed to the next prompt:

BRANCH>

BRANCH> responds to the following commands.

TOTALS - Same as above.

EXIT - Same as above.

AUTOMATIC - Computer responds
START WITH WHICH LEVEL?
User responds with an integer, n.
The computer asks,
POST-EDIT?
User responds "yes" or "no", the consequence of which will be explained momentarily. The

branching algorithm is now invoked which determines connectivity on the basis of window overlap. Branching and triangulation are also handled automatically. If POST-EDITing was refused, the entire data file is thus triangulated - with no gaurantees. If POST-EDITing was requested, each triangulated contour pair is displayed for approval as the computer asks,

CHANGES?

Two things may be in error - the triangulation itself, or the choice of loops involved in branching. If either of these errors necessitate change, answer "yes". The computer then asks,

CHANGE CONNECTIVITY?

If the branching was at fault, answer "yes". This has the effect of changing from AUTOMATIC to MANUAL mode, and the user is free to manually control the branching. If only the triangulation was bad, answer "no". This moves the program counter to the TRIANGULATE> prompt, so the user can guide triangulation manually.

If no changes at all are needed, a carriage return pronounces acceptance and the algorithm procedes to the next contour pair.

WARP - Computer responds

MAX. WARP ANGLE:

Here the user sets the Maximum Warp Angle described in chapter 5. Default is 45 degrees.

INSPECT - Computer responds

START WITH WHICH LEVEL?

After which the user responds with an integer, n. The tektronix scope then displays the contour line(s) of level n, labelled with global loop numbers. A carriage return will cause level n+1 to be displayed; a "control-Z" (end of data) returns control to BRANCH>.

MANUAL - Again, the computer asks,

START WITH WHICH LEVEL?

The user responds with an integer,n. This initiates branching and triangulation. Each pair of neighboring contour levels is considered respectively ranging from n to the last level read in. All contours from the two relevant levels are displayed, and identified by a number followed by T(top) or B(bottom). The user selects from these loops the ones which belong together for triangulation. (This need only be used for complex cases of branching. Normally, the AUTOMATIC command handles branching adequately.) The computer asks:

BOTTOM LOOP(S) (LOCAL):

Here, the user enters the loop numbers of all the loops that are to participate in triangulation. The loop numbers are delimited by carriage returns. The series is terminated by by a double carriage return, after which the computer asks:

TOP LOOP(S) (LOCAL):

The same procedure is again followed in inputting the top loops. This sequence of loop numbers is loaded into an array called Loopstack, which is passed to SUBROUTINE QUAD for triangulation. If there are more than one loop on either top or bottom (i.e. branching) QUAD displays all the loops and inquires of the user how to assemble them into one loop by asking

ENTER LOOP SEQUENCE:

The user now enters the sequence in which the loops are to be re-constructed (as explained in chapter 3). If not all loops are intended to be included, the sequence may be terminated with a control-Z. Next, the computer asks how to interconnect the loops by typing: (where the loop sequence is i,j,k)

LiP?

Enter the node on loop i to be joined to loop j.

LjP?

Enter the node on loop j to be joined to loop i.

etc.

When this series of questions is answered, the routine re-arranges the branching loops into one loop (as explained in chapter 3), and proceeds to the final command prompt of TRIANGULATE>. Commands for that prompt are discussed later.

SINGLE - This command performs a function similar to that of MANUAL. The difference is that global loop numbers are used instead of local. This enables any loop(s) to be defined as top or bottom. The SINGLE command is used only in irregular situations involving concentric loops. Execution procedes as in MANUAL.

CAP - CAP invokes an algorithm to form a pyramidal cap on a specified loop. The computer asks:

GLOBAL LOOP NUMBER:
To find out what the global loop number is (in case the user doesn't know) use the INSPECT command. This should be done prior to invoking the CAP command.

WARP, CAP, INSPECT, and TOTALS return to the BRANCH> prompt. AUTOMATIC returns to the READ> prompt after triangulating all available contours.

The final prompt, TRIANGULATE>, is only encountered during the BRANCH command, or possibly during the AUTOMATIC/POST-EDIT command (if triangulation is faulty).

TRIANGULATE> responds to these commands:

AUTOMATIC - Same as the AUTOMATIC command in the BRANCH> menu.

DENSITY - As default, the program labels every other node for identification. If this labelling is too dense to decipher (or too sparce), the DENSITY command provides help.
The computer asks,
IND3= , where IND3 is the 3rd DO LOOP parameter in the labelling loop.

TOTALS - Same as before.

PART - The MOVIE.BYU graphics package enables panels to be grouped together as parts. Each part can then be addressed seperately with its own set of display parameters. Here, the PART command funnels all succeeding panels into a specified part number. The computer asks:
CHANGE PART FROM n TO :
where n is the currently assigned part number. Initial part use must proceed in numeric order. That is, part 2 must be used before part 3, etc. However, once used, there is no restriction on future use, i.e. the part number could be changed from 5 to 2, for example. The algorithm safegaurds these rules, and notifies the user of violations.

UNMAP - Sets a flag that causes the mapping algorithm to be bypassed, so contours are considered in their initial orientation for triangulation.

MAP -	Opposite of UNMAP. Default is MAP.
ONE -	Permits the user to manually define a single panel from any 3 (or 4) nodes from the contour pair.
INTERACTIVE -	Permits interactive guidance of the triangulation by requesting limits for both loops between which to triangulate. The computer types TOP: n-m where n is the last node to be triangulated and m is the number of nodes on the top contour. (n and m are merely stated for clarity). The user responds with two integers delimited by a comma. The first delimiter is generally n, and the second is always $\leq$ m. Delimiters are also requested for the bottom loop. Triangulation occurs within the prescribed limits and is displayed for approval. If approval is denied, only the newest panels are erased, and the user is invited to try again. A control-Z returns control to TRIANGULATE>.
ERASE -	Erases all panels from the present pair of contours.
RENUMBER -	Triangulation is facilitated if node 1 on the top loop neighbors node 1 on the bottom loop. RENUMBER enables this by asking NEW #1 NODE (TOP CONTOUR) The renumbered contours are then displayed. This renumbering must occur before any triangulation. AUTOMATIC automatically renumbers before triangulating.
CONTINUE -	Returns control to the main program.

MOVIE.BYU Mailing List

Department of the Air Force
Air Force Wright Aeronautical Lab.
Air Force Flight Dynamics Lab.
Wright-Patterson Air Force Base,
Ohio 45433
ATTN: FBRA(Mr. L. Bernier/
513-255-5689)

* Ms. Betty Cuthill
Department of the Navy
Naval Ship Research and Dev. Ctr.
Bethesda, MD 20034

* Prof. Dwight T. Davy
Department of Engineering Mech.
The University of Nebraska/Lincoln
Lincoln, NB 68588

Mr. Howard Eagle
Advanced Struc. Anal. Rm. 2054-CCF#9
General Electric Company
P. O. Box 7560
Philadelphia, PA 19101

* Professor Larry J. Feeser
Dept. of Civil Engineering
Rensselaer Polytechnica Inst.
Troy, NY 12181

* Mr. Norman L. Firkins
Gibbs and Hill, INc.
8420 West Dodge Road
Omaha, NB 68114

* Professor Movses J. Kaldjian
Department of Civil Engineering
The University of Michigan
Ann Arbor, MI 48104

* Mr. R. E. Love
Data Processing
Bechtel International Corp.
P. O. Box 3965
San Francisco, CA 94119

* Mr. B. W. Maddock
Engineering Services
Sperry Univac Computer Systems
P. O. Box 500 M.S. B216M
Township Line and Jolly Road
Blue Bell, PA 19422

* Professor D. W. Murray
Department of Civil Engineering
The University of Alberta
Edmonton, Alberta
CANADA T6G 2G7

* Mr. Mark F. Nelson
Engineering Mechanics Dept.
Research Laboratories
General Motors Corporation
General Motors Technical Ctr.
Warren, MI 48090

* Dr. C. J. Parekh
Engineering Mech. Research Corp.
Green-Lincoln Office Plaza
25900 Greenfield Road
Suite 305
Oak Park, MI 48237

Mr. R. A. Ridha
The Firestone Tire and Rubber Co.
1200 Firestone Parkway
Akron, OH 44317

* Mr. R. B. Smith
Bettis Atomic Power Laboratory
Westinghouse Electric Corporation
Box 79
West Mifflin, PA 15122

* Professor Ronald Stearman
Dept. of Aerospace Engineering
and Engineering Mech.
The University of Texas
Austin, TX 78712

* Dr. Eric J. Stevens
Electromechanical Sys. Eng.
Mass. Inst. of Tech.
Lincoln Laboratory
Lexington, Mass. 02173

* Prof. Thein Wah
Civil and Mech. Engineering
Texas A & I University
Kingsville, TX 78363

* Indicates that they have a copy of the system.

* Mr. Jim Jones
Automatic Data Processing Center
U.S. Army Engineers Waterways
Experiment Station
P. O. Box 631
Vicksburg, MI 35180

Mr. Rauli Uitto
Lockheed Electronics
Rocket Lab
Edwards Air Force Base
Edwards, CA 93523

* Mr. Joseph A. Curreri

P.O.Box 8217
Southwest Station
Washington, D.C. 20024

* Mr. Larry Hubble
ESL Incorporated
495 Java Drive
Sunnyvale, CA 94086

Professor Jim Linford
MIU and MERU University
6446 Seblisberg
SWITZERLAND

Mr. Harold Noffke
AFAL/RWM - 5
WPAFB, Ohio 45433

* Mr. Dan Henderson
Code 74
NISC
4301 Suitland Road
Washington, D.C. 20390

* Mr. Mal Carey
University of Maine at Orono
Social Science Research Institute
164 College Avenue
Orono, ME 04473

* Mr. H. Merryweather
5 Manor Drive
Fenstanton
Huntington, Cambs PE 18 9QZ
ENGLAND

* Mr. Jeffrey Kulick, Asst. Prof.
Department of Computing and
Information Science
Queens University
Kingston, CANADA K7L 3N6

* Mr. Michael A. Tamny
Code 5365T
Naval Research Laboratory
Washington, D.C. 20375

Mr. Edgar T. Lynk
Corporate Research and Development
General Electric
P. O. Box 8
Schenectady, NY 12301

* Mr. E. Gamble
Chemistry Department
The Research Foundation
SUNY at Stony Brook
Stony Brook, NY 11794

Mr. Peter R. Paradis
Computer Services Directorate
Naval Air Test Center
Patuxent River, MD 20670

Mr. Warren Cunningham
2437 Jefferson Avenue
Berkeley, CA 94703

Mr. A. Stein
IBM Corporation
P. O. Box 218
Yorktown Heights, NY 10598

* Mr. Larry McCleary
Dept. of the Navy
Naval Undersea Center
Code 302
San Diego, California 92132

* Dr. Bill Cook
R-4 MS 561
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87544

* Mr. Kennith J. Carter
Eimco Corporation
669 W. 200 S.
Salt Lake City, Utah 84100

* Mr. Morten Zachrisen
RUNIT Computing Center
University of Trondheim
N-7034 Trondheim - NTH
NORWAY

* Dr. O. Wang
P. O. Box 270, MC 8840
John Deere Waterloo Product
Engineering Center
Waterloo, IW 50701

Mr. C. J. Washam
Control Data Corporation
Arden Hills Operations
4201 North Lexington Avenue
St. Paul, MN 55112

Ms. Sandra Lapietra
Dravo Corporation
One Oliver Plaza
System Section
Pittsburg, PA 15222

* Mr. Malcolm Beyer
Command Control and Communications
Corporation
P. O. Box 242
San Pedro, CA 90733

Mr. David F. Rogers
United States Naval Academy
Annapolis, Maryland 21402

* Mr. Phillip S. Mittleman,
President
Mathematical Applications Group,
Inc.
3 Westchester Plaza
Elmsford, NY 10523

* Mrs. C. Conley
Teknica Resource Development Ltd.
Suite 412
339 - 6th Avenue S.W.
Calgary, Alberta T2P 0R8
CANADA

* Dr. G. Maymon
Ministry of Defense
Scientific Department
Haifa 31000
P.O. Box 2250
ISRAEL

* Dr. Steven E. Benzley
Division 1542
Sandia Corporation
Albuquerque,, NM 87115

* Mr. Jay W. Wiley
Bechtel Power Corporation
12400 East Imperial Highway
Norwalk, CA 90650

* Lt. Col. Joseph D. Morgan, III
DFCEM
USAF Academy, CO 80840

* Mr. Richard G. Gauthier
General Dynamics, Department 443
Electric Boat Division
Eastern Point Road
Groton, CT 06340

* Professor Horst Nowacki
Fachgebiet Schiffsentwurf
Institute Fur Schiffstechnik
Technische Universitat Berlin
Salzufer 17/19
1000 Berlin 10
WEST GERMANY

* Mr. Lawrence K. Yu
Research Division
General Tire and Rubber Co.
2990 Gilchrist Road
Akron, Ohio 44305

* Dr. Harry C. Andrews
Image Processing Institute
Powell Hall
University of Southern California
University Park, CA 90007

* Prof. Charles Beadle
Dept. of Mechanical Engineering
College of Engineering
University of California
Davis, CA 94616

* Mr. David W. Princehouse
Applied Physics Laboratory
University of Washington
1013 Northeast Fortieth Street
Seattle, WA 98195

* Mr. Art Carlson
Code EM
Naval Underwater Systems Center
New London, CT 06320

* Dr. Robert C. Leif
Papanicolaou Cancer Research
Institute at Miami Inc.
Box 236188
Miami, FL 33123

✻ Mr. Duane Soland
Senior Principal Research Engr.
Honeywell Systems & Research Cntr.
2600 Ridgway Parkway
Minneapolis, MN 55413

Mr. Ken Knepper
Tektronix, INc.
P. O. Box 500
Delivery Station 60-287
Beaverton, OR 97077

✻ Mr. Charles R. Patton
SYNERGY
Box 676
Stanton, CA 90680

✻ Mr. Dennis Conroy
Kaman Sciences Corporation
P. O. Box 7463
Colorado Springs, CO 80933

✻ Mr. Charles Zonca
Ford Motor Company
Car Engineering Group
P.O. Box 2053
Bldg. 4--Rm 115A
Dearborn, MI 48121

Ms. Carley Ward
Structures Division
Code L51
Department of the Navy
Civil Engineering Laboratory, NCBC
Port Hueneme, CA 93041

✻ Mr. Serge Chicoine
Computer Center
Universite de Montreal
Case Postale 6128
Montreal 101, Canada

✻ Mr. Henry Lee
AMOCO Production Co.
4502 East 41st Street
P.O. Box 591
Tulsa, OK 74102

✻ Mr. Charles V. Smith
Pittsburgh Des Moines Steel Co.
Neville Island
Pittsburgh, PA 15225

✻ Dr. James J. Soltis
University of Windsor
Department of Electrical Engineering
Windsor, Ontario N9B 3P4

✻ Mr. R. H. Batterman
Alcoa Laboratories
Alcoa Technical Center
Alcoa Center, PA 15069

✻ Mr. Ron Schmitz
Sperry Support Service
716 Arcadia Cir.
Huntsville, Alabama 35801

✻ Mr. Herbert E. Zellnik
United Engineers & Constructors Inc.
30 South Seventeenth Street
Philadelphia, PA 19101

✻ Mr. Robert Pine
Applied Computer Technology Inc.
17 Norway Court
Newark, DE 19711

✻ Professor Jerald A. Griess
Dept of Industrial Education
Eastern Michigan University
Ypsilante, MI 48197

Professor Del Coates
Industrial Design Department
Center for Creative Studies
245 East Kirby
Detroit, MI 48202

Mr. George S. Host
Arthur G. McKee & Co.
Cleveland, OH 44131

✻ Mr. T.G. Creedun
Theodore G. Creedun & Co.
1011 East Tudor Road
Suite 280
Anchorage, AK 99503

✻ Dr. Donald P. Greenberg
Computer Graphics Laboratory
120 Rand Hall
Cornell University
Ithica, NY 14853

* Mr. Sam Bosch, MR 2-4
Digital Equipment Corporation
1 Iron Way
Marlboro, MA 01752

* Mr. Roland E. Johnson
The Proctor and Gamble Co.
P.O. Box 599
Cincinnati, OH 45201

* Mr. Allan R. Holdridge
Ex-Cell-O Corporation
850 Ladd Road
Walled Lake, MI 48088

* Mr. Vincent J. Vitagliano
IBM Corporation
360 Hamilton Avenue
White Plains, New York 10601

* Mr. Robert A. Wells
Engineering Systems Manager
Multisystems, Incorporated
1050 Massachusetts Avenue
Cambridge, MA 02138

* Dr. Sam Zentman
Engineering Computer Center
American Motors Corporation
14250 Plymouth Road
Detroit, Michigan 48232

* Mr. R. L. Wallace
International Harvester Company
Melrose Park Plant
1040 W. North Avenue
Melrose Park, IL 60160

* Mr. Richard Wiersma
Wayne State University
Computing and Data Processing
Detroit, Michigan 48202

* Professor Lawrence E. Wilborn
Mining Engineering
Michigan Technological University
Houghton, Michigan 49931

* Mr. R.A. Froats
Proctor and Redfern Ltd.
75 Eclinton Ave. East
Toronto, Ontario, Canada
M4P 1H3

* Professor David C. Anderson
School of Mechanical Engineering
Room 122
Purdue University
West Lafayette, Indiana 47907

* Mr. Donald H. Carter
Smith, Hinchman, Grylls Associates Inc.
455 West Fort Street
Detroit, Michigan 48226

* Mr. Andrew J. Blum
Herman Blum Consulting Engineers
1015 Elm Street at Griffin
Dallas, Texas 75202

* Dr. Lee J. Ovenshire
Office of Vehicle Systems Research
Structures Research Division
Research and Development
U.S. Department of Transportation
National Highway Traffic Safety Administ
Washington, D.C. 20590

* Clifton E. Jackson, Jr.
Code 720.2
Goddard Space Flight Center
Greenbelt, MD 20771

* Mr. Larry Howson
General Motors Transportation System Div
12 Mile and Mound Road
Warren, Michigan 48090

* Ms. Joanne Chiocchio
Stone and Webster Engineering Corporatio
P. O. Box 2325
Boston, MA 02107

* John Gareiss
Schmidt Garden and Erikson
104 South Michigan
Chicago, IL 60603

* Mr. Frank D. Smith
Arthur G. McKee and Company
6200 Oaktree Blvd.
Independence, Ohio 44131

* Prof. R. Phillips
Gas Dynamics Labs.
218 Aerospace Engineering-North Campus
The University of Michigan
Ann Arbor, Michigan 48109

*Mr. Liu Hsun
Bio-Physics Science Unit
MAYO Clinic
Rochester, Minnesota 55901

*David Zeltner
150 N. Pioneer
Ashland, Oregon 97520

*Richard P. Wehrle
International Harvester Co.
2911 Meyer Road
Fort Wayne, Indiana 46803

Dr. Norman Wolcott
Computer Service Division
Room A221, Administration Building
National Bureau of Standards
Washington, D.C. 20234

Steven Gregory
Graduate School of Architecture
University of Utah
Salt Lake City, Utah 84112

*Mr. Eamon Barrett
ESL, Incorporated
495 Java Drive
Sunnyvale, CA 94086

*Chrysler Corporation
1250 Product Development Functions
12800 Oakland Ave.
Highland Park, MI 48203
ATTN: F. H. Holzhauer
Supervisor Engineering Systems
Development

*Professor Mike Wilkinson
Box 6275
Louisiana Tech University
Ruston, LA 71270

*Mr. Joseph J. Gallagher
Raytheon Co.
Hartwell Road
Bedford, MA 01730

*Mr. William Tarr
Illinois Department of Transportation
Room 107
2300 South Dirksen Parkway
Springfield, Illinois 62764

Mr. Michael Neighbors
B.K. Dynamics
200 West Court Square
Huntsville, Alabama, 35801

Mr. David E. Weisberg
District Sales Manager
Tektronix, Inc.
14 Inverness Drive East
Englewood, Colorado 80110

Mr. Bruce E. Brown
595 S. 875 W.
Woods Cross, Utah 84087

*Professor Michael B. Stephenson
Dept. of Civil Engineering
and Engineering Mechanics
The University of Arizona
Tucson, Arizona 85721

Will Hendricks
5736 W. 81st Terrace
Prairie Village, Kansas 66208

*Mr. Christian Rouxel
Institut de Recherches
de la Construction Navale
47, rue de Monceau
75008 Paris
France

*Dr. John Hunt and Dr. Dick Larder
Lawrence Livermore Laboratories
University of California
Box 808
Livermore, California 84550

*Mr. W. C. Stoddart
Oak Ridge National Laboratory
MS 14, Bldg. 9204-1
Oak Ridge, Tennessee 37830

*R.D. Erickson
Computer Services Division
Michigan Dept. State Highways and Trans.
P.O. Box 30050
Lansing, Michigan 48909

*Mr. W. M. Lambert
Control Data Corporation
23815 Northwestern Highway
Southfield, Michigan 48075

*Mr. Bill Kunkel
Westinghouse Electric Corp.
2040 Ardmore Blvd.
Pittsburg, Pennsylvania 15221

*Mr. Frank R. Johnson, Jr.
Structures Division
Civil Engineering Laboratory
Naval Construction Battalion Center
Port Hueneme, CA 93043

Mr. I. W. Dingwell
Arthur D. Little, Inc.
Acorn Park
Cambridge, MA 02140

*Betty Joe Armstead
MS 86-2
Louis Research Center, NASA
21000 Brookpark Road
Cleveland, Ohio 44135

*Dr. Geoffrey A Butlin
Kongsberg-Ikoss Consultants A/S
Nils Hansens Vei 2
Oslo 6 NORWAY

*Mr. Bob Guenzler
A. G. and G.
550 2nd Street
Idaho Falls, Idaho 83401

Yngvar Lundh
Norwegian Defence Research Est.
P. O. Box 25
2007 Kjeller
NORWAY

*Jack L. Winger
AFATL/DLYV
Elgin AFB, FL 32542

Greg Corson
19141 Summers Dr.
South Bend, IN 46637

Peter J. Feil and Greg Guthrie
Purdue University
A&ES Building Rm. 115
West Lafayette, IN 47907

Dan Eesley
Amdahl Corporation
1250 Arques Ave.
Sunnyvale, California 94086

Merrell R Jones
Computer Center
Southern Utah State College
Cedar City, Utah 84720

Gerald W. Call
CALL/SYSTEMS, Inc.
Box 8155
Portland, Oregon 97207

John J. Cornman
Ranier National Bank
1321 2nd Ave. - 7th Floor - I. S.
Seattle, WA 98101

Charles Eastman
Dept. of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

Tim Zimmerlin
Technology Service Corporation
2811 Wilshire Blvd.
Santa Monica, CA 90404

Joe Capowski
Dept. of Physiology
University of North Carolina
Chapel Hill, NC 27514

*Phil Sherrod
Vanderbilt University
Box 1577, Station B
Nashville, TN 37235

Dennis L. Fung
Department of Anethesiology
4301 X Street, Room 253
Sacramento, CA 95817

John Frink, WA 32
Naval Surface Weapons Center
White Oak Laboratory
Silver Spring, Maryland 20910

Harry Gough
180 Old Enterprise Road
Upper Marlboro, MD 20870

Dick Elkins
Boeing Company
MS 8H - I3
Box 3999
Seattle, WA 98124

Matti Loikkanen
Control Data Canada, Ltd.
4715 1st Street, S. W.
Calgary, Alberta
Canada, T2G 0A1

✱ William S Rhode
283 Medical Sciences
University of Wisconsin
Madison, WI 53706

Pat Landau
Single Buoy Moorings
57 Rue Grimaldi
Principality of Monaco, M. C.
EUROPE

✱ David A Pensak
Central Research Department
Du Pont Experimental Station
Wilmington, Delaware 19898

Dr. Richard Gordon
Bldg. 36, Room 4D28
Dept. of Health, Educ. and Welfare
National Institute of Health
Bethesda, MD 20014

A C Landsburg
Maritime Administration
307 Williamsburg
Silver Spring, MD 20901

Dave Fuhrer
Computer Center
Oregon State University
Corvallis, Oregon 97330

Louis D Grey
Perkins-Elmer Corporation
100 Wooster Heights Road
Danbury, CT 06810

James Steinberg /23
Transportation Systems Center
Kendall Square
Cambridge, MA 02148

David L Libby
University Computer Center
The University of Iowa
Iowa City, Iowa 52242

Robert Dosen
Fermi Lab.
Box 500 CL-8E
Batavia, IL 60510

Robert C Adamson
Chevron USA Inc.
1111 Tulane Ave.
New Orleans, LA 70112

✱ Richard H Blocher
Box 1152
University Computing Facilities
Washington University
Saint Louis, Missouri 63130

Richard E Lampe
WTA Computer Services, Inc.
2357 59th Street
St. Louis, Missouri 63110

✱ Sam Shum
Association of American Railroads
3140 S. Federal St.
Chicago, IL 60616

C H Roth
Electrical Engineering Dept.
University of Texas
Austin, Texas 78712

Jo Ann Baughman
Computer Center
Oregon State University
Corvallis, Oregon 97331

George E Carroll
Marine Sciences Research Center
SUNY at Stony Brook
Stony Brook, New York 11794

George T Hawkins
USACAA
8120 Woodmont Ave.
Bethesda, MD 20014

Kenneth W Speaker
Dept. of Zoology
University of Texas
Austin, Texas 78712

Charles R Watson
Biemetrics Center
Statistics Section
Battelle Pacific Northwest Laboratories
Battelle Boulevard
Richland, Washington 99352

* James W Dow
Southern Company Services
Perimeter Center East
Box 720071
Atlanta, Georga 30346

* Robert K Clark
Applied Mathematics
Argonne National Laboratory
Argonne, IL 60439

Malcolm C Babb
Tennessee Valley Authority
262 401 Building
Chattanooga, TN 37401

* E L Hall
Department of Electrical Engineering
The University of Tennessee
Knoxville, Tennessee 37916

Holly B Peirce
Production Automation Project
University of Rochester
Rochester, New York 14627

William Wisse
CompuServe Incorporated
5000 Arlington Ctr. Blvd.
Columbus, Ohio 43220

Larry Nolan
McDonnell Douglas Automation Co.
Box 516, Dept K251
St. Louis, MO 63166

Harold P Spahr
Sandia Labs., Division 1336
Box 5800
Albuquerque, New Mexico 87123

R L Fellows
Oak Ridge National Laboratory
Box X, Bldg. 5505
Oak Ridge, TN 37830

Dr. Howard F Slater
USACOMISA
Attn: CCD-SE
Ft. Huachuca, Arizona 85613

Doug Richardson
CADIG Group, Div. of Eng. and Weapons
US Naval Academy
Annapolis, MD 21402

Christos I Yessios
4367 Mumford Dr.
Columbus, Ohio 43220

Francis T Meissner
NASA Langley Research Center
MS 125
Hampton VA 23665

Dr. Frederick P Brooks
Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27514

* Gary N Boughan
Vitro Labs. STB-13 10-1002
14000 Georgia Ave.
Silver Springs, MD 20910

H Pierpont
GCIA, Inc.
258 Rochester Road
Mobile, AL 36608

Brian J Winkel
Dapartment of Mathematics
Albion, MI 49224

Loren Carpenter
Boeing Computer Services
MS 36-01
Box 24346
Seattle, Washington 98124

* Gary R Montry
KMS Fusion Inc.
Box 1567
Ann Arbor, MI 48104

* David J Rahnis
Autographics, Inc.
Village of Cross Keys #206
Baltimore, MD 21210

Dr. Peter P Dinyovszky
Sikorsky Aircraft
North Main Street
Stratford, CT 06602

* B M Bardin
Indiana University Cyclotron Facility
Milo B Sampson Lane
Bloomington, IN 47401

Dan R Schenck
Cities Service Company
Room 2131
Box 300
Tulsa, Oklahoma 74102

Julie Watts
Environmental Sciences Division, ORNL
Box X Oak Ridge, TN 37830

Dr. Caleb A George
General Electric Co.
Aerospace Electronic Systems Department
Box 524 MD 348
Utica, NY 13503

Dennis Wilson
Computer Sciences Corporation
1528 East Missouri Ave.
Pheonix, Arizona 85014

Thomas E Shields
Software Resources
2715 Bissonnet, Suite 212
Houston, Texas 77005

Professor Leonard Soltzberg
Department of Chemistry
Simmons College
300 Fenway
Boston, MA 02115

*McKay Anderson
Hercules Incorporated
Box 98
Magna, Utah 84044

Karl H Ryden
Health Sciences Computing Facility
UCLA
Los Angeles, CA 90024

Roy Law
Naval Weapons Center
Code 3107
China Lake, CA 93555

John P. Long
Computer Graphics, Division 2644
Sandia Laboratories
Albuquerque, NM 87115

Chuck Weger
CUNY/UCC
555 West 57th Street
New York, NY 10019

*S. Nils Straatveit
Code 315
Naval Underwater Systems Center
New London, CT 06320

Dr. C. S. Bauer
IEMS Dept.
Florida Technological University
Box 25000
Orlando, FL 32816

L. G. Clark, Jr.
Monsanto - MISD - G3EA
800 N. Lindbergh
St. Louis, MO 63166

Kathleen N Fischer
Computer Sciences Division
Union Carbide Corporation
Box X
Oak Ridge, Tennessee 37830

R. A. Lovestedt
Boeing CAD R&D
Box 3707 MS 36-01
Seattle, WA 98124

John H Leary
Avondale Shipyards, Inc.
Box 50280
New Orleans, Louisiana 70115

John R. Norris Jr.
Rockwell International
3117 Washington Pike
Bridgeville, PA 15017

Greg Bettice
Naval Avionics Facility
6000 East 21st Street D/822
Indianapolis, IN 46218

Richard A Stahle
Pacific Missile Test Center
Code 3451-1
Systems Analysis Section
Point Magu, CA 93042

Gordon L Herald
USA Human Engineering Laboratory
Building 520
Aberdeen Proving Grounds, MD 21005

Seigfried W. Pudell
Clark Equipment
Box 547
Benton Harbor, MI 49022

Bruce E. Koopman
Sandia Laboratories
Division 8323
Livermore, CA 94550

*C. S. Smyth
US Dept of Commerce
NOAA/ERL/PMEL
3711-15th NE
Seattle, WA 98105

M. Alan Bishop
DBA Systems, Inc.
P. O. Drawer 550
Melbourne, FL 32901

Donald W Holder
DRSMI-TDD (aeroballistics Dir.)
Redstone Arsenal, AL 35809

USA-CEEIA-CONUS
Attn: CCCN-AT (bldg. 117)
Ft. Ritchie, MD 21719

Lou Perazzoli
Digital Equipment Corporation
5900 Princess Garden Parkway
Lanham, MD 20801

*Linda P. Dodge
Shell Development Co.
Computer Science Section
Box 481
Houston, Texas 77001

Awni Boutros
Evasco Services
2 Rector Street, Room 1549
New York, NY 10006

*Walter H Niehoff
IBM Corp. - Dept. T71
Box 6
Endicott, NY 13760

*D R Sudweeks
General Dynamics/Convair Division
Box 80847, Mail Zone 627-00
San Diego, CA 92138

Steven H. Weiss
Gull Lake Laboratories
3700 East Gull Lake Drive
Hickory Corners, MI 49060

Leonard J Feldman
Compusystem
1308 6th Ave.
Beaver Falls, PA 15010

*Dr. Randolph H Levine
Harvard College Observatory
60 Garden Street
Cambridge, MA 02138

Ned Buchman
General Electric Co.
5030 Herzel Place
Beltsville, Maryland 20705

Robert Mallary
University of Massachusetts, Amherst
Box 48
Conway, MA 01341

J F Novak
Novacolor, Inc.
599 North York Road
Elmhurst, IL 60126

C R Garthwaite
University of Washington
Pathology SM - 30
Seattle, WA 98195

*Dr. Roger S Pressman
University of Bridgeport
Technology Bldg
Bridgeport, CT 06602

Ron W. Brittian
Texas Instruments Inc.
13500 North Central Expressway
M/S 992
Dallas, Texas 75222

Robert B Thornhill, Director
Division of Engineering Technology
Wayne State University
Detroit, MI 48202

Bruce Kraemer
Manager, Time Sharing and Tech. Services
Foxboro Company
Cocasset Bldg., Mechanic Street
Foxboro, MA 02035

Stephen J Bartholet
Odetics Inc.
1859 South Manchester Avenue
Anaheim, CA 92802

Al Cinque
David Taylor Naval Ship R&D Center
Code 1854
Bethesda, Maryland 20084

W. Huber
GENRAD
2855 Bowers
Santa Clara, CA 95051

Jack J. Staller
Graphics Automation Co.
661 Washington St. Suite 302
Norwood, MA 02062

* John R. Skeen, Jr.
TRW DSSG, Design Analysis Center
One Space Park, M3/2042
Redondo Beach, CA 90278

* M W Vannier
Wenner Gren Laboratory
University of Kentucky
Lexington, KY 40506

Tommy G Dearmond
University of Southern Mississippi
Box 286 Southern Station
Hattiesburg, MS 39401

Wm. Johnston
Lawrence Berkeley Laboratory
Bldg. 50B, Room 2245
Berkeley, CA 94720

Dr. Julian Kately, Jr.
Michigan State University
Computer Laboratory
East Lansing, MI 48824

Richard A Pollak
Academic Computing Center
United States Naval Academy
Annapolis, Maryland 21402

* Mark Jaffe
Vector General, Inc.
21300 Oxnard street
Woodland Hills, CA 91367

* S. Russell Herron
Exxon Production Research Co.
Box 2189
Houston, TX 77001

W B Boyer
Sandia Lab
Box 5800
Albuquerque, NM 87165

R F Mills
Division of State and Regional Planning
329 W State Street
Trenton, NJ 08625

Dr. B C Platt
University of Dayton
AFWL/LREm Bldg. 401
Kirtland AFB, NM 87117

Major John J. Russell
Dept. of Civil Engr. Mech. and Mat'ls.
(DFCEM)
USAF Academy, CO 80840

Fredrick T. Fink
College of Engineering
Michigan State University
East Lansing, MI 48824

* H. Alpaugh M/S 27
C. S. Draper Laboratory, Inc.
555 Technology Square
Cambridge, MA 02139

Dr. J. E. Adams
Stephen F. Austin State University
Box 3040 SFA
Nacogdoches, TX 75962

Ronald Janetzke
NASA-GSFC
Code 626
Greenbelt, MD 20771

James W Guinan
162 Woodbine Avenue
Staten Island, New York 10314

Earl Barnes
V. A. Hospital
Nuclear Medicine Svc.
Hines, IL 60141

Lawrence N. Shaw
Palace of Arts and Sciences
The Exploratorium
3601 Lyon Street
San Francisco, CA 94123

John M. Baker
Union Carbide Corporation
Nuclear Division
Box P
Oak Ridge, TN 37830

Bruce Bullock
Hughes Research Laboratory
3011 Malibu Canyon Road
Malibu, CA 90265

Hsun Kao Liu
Computer Science Dept. SUNYAB
4226 Ridge Lea Road
Amherst, NY 14226

Raymond R. Bober
Ebasco Services Inc.
2 Rector Street, Rm 1546
New York, NY 10006

* Rex Tracy
Colorado State University
Electrical Engineering Dept.
Ft. Collins, CO 80523

* John M Crowell
Los Alamos Scientific Laboratory
Box 1663, MS 888
Los Alamos, NM 87545

Jerome Schuh et al.
MESOMET, Inc.
3415 University Ave. N. E.
Minneapolis, NM 55414

Chit F Chung
Weidlinger Assoc.
110 E 59th Street
New York, NY 10022

Edward A Hoar
Tektronix, Inc.
Box 500
Beaverton, OR 97077

Ralph A Kinney
Electrical Engineering Dept.
Lousiana State University
Baton Rouge, LA 70803

Gary W Meyer
Bell Laboratories
Rm 2B218A, Crawford Corner Road
Holmdel, NJ 07733

Frank D Harrison
Maricopa County Highway Dept.
3325 West Durango
Pheonix, AZ 85009

T B Priest, Jr.
Georgia Institute of Technology
500 Northside Circle
Atlanta, GA 30309

James L Boyd
USAFETAC/CBC
Scott AFB, IL 62225

* Milt Waxman
Hughes Aircraft
Building 6 Mail Stop D-114
Culver City, CA 90230

* Barbara Ringers
Dept. of the Army - Ballistic Research Labs.
Aberdeen Proving Ground
Aberdeen, Maryland 21005

* Loren Carpenter
Boeing Computer Services
Mail Stop 36-01
Box 24346
Seattle, Washington 98124

David C Allen
Sperry-Univac
Building E
322 North 22nd West
Salt Lake City, Utah 84116

M. W. Krimmer
US Army Rocky Mountain Arsenal
Attn:SARRM-IRD
Denver, CO 80240

Harold W. Krauss
Chevrolet Engineering Center
30003 Van Dyke, Rm. C-2-232
Warren, MI 48090

Robert J. LaPine
Chevrolet Engineering Center
30003 Van Dyke, Rm C-2-232
Warren, MI 48090

Issevar Saknussem
5204 12th N. E.
Seattle, WAshington 98105

Andy Bouillot
Occidental Petroleum Corporation
Grand Island Data Center
Box 8
Niagara Falls, NY 14302

John H Myers
TimeWare
530 Lytton Avenue - Suite 370
Palo Alto, CA 94301

*Harland E Alpaugh
Draper Laboratory
555 Technology Square
Cambridge, MA 02139

*Harvey Sellner
Perkin-Elmer Corporation
100 Wooster Heights Road
Danbury, Conn. 06810

Paul Hughett (Consultant)
P. O. Box 60
Palo Alto, CA 94302

Robert B. Weinberg
Physics Department - Barton Hall
Temple University
Philadelphia, PA 19122

Robert F. Bachert
6570 AMRL/HEB
Wright-Patterson AFB
Dayton, Ohio 45433

John H. Fulton, Computing Center
N.C. State University
P. O. Box 5445
Raleigh, N.C. 27607

Richard Cicone
Environmental Research Institute
of Michigan
PO Box 618
Ann Arbor, Michigan 48107

G. D. Kuhn, DB32
Naval Avionics Facility
6000 E. 21st Street
Indianapolis, Indiana 46218

Harris A. Meyers
Optimum Systems Inc.
5615 Fishers Ln.
Rickville, MD 20852

David S. Backer
536 Main Street, #1
Amherst, Mass. 01002

Michael J. Frisch
University Computing Center
University of Minnesota
227 Experimental Engineering Bldg.
Minneapolis, MN 55455

Nelson A Logan
Mail Code B14
Lockheed Electronics Company
16811 El Camino Real
Houston, TX 77058

*Scott Evernden
Applicon, Inc.
154 Middlesex Tnpk
Burlington, MA 01803

Bill Krueger
Space Science and Engineering Cent
The University of Wisconsin
1225 West Dayton Street
Madison, Wisconsin 53706

*Richard Eppes, Jr.
DRDMI-TLA, Dldg 5400
US Army Missile Research and Development
Redstone Arsenal, AL 35809

Jim Foley
Bureau of the Census-SSD
Washington, DC 20233

Hatim J Kanorwala
Sun Shipbuilding and Dry Dock Co.
Chester, PA 19013

Chris Petersen
Renaissance Systems
11760 Sorrento Valley Rd. Suite M
San Diego, CA 92121

*Dick Bartlein
2468 N. Frederick
Milwaukee, WI 53211

*Bob Guenzler
E. G. & G. Idaho, Inc.
Box 1625
Idaho Falls, Idaho 83401

*Albert Allen
Academic Computing
Southern Illinois University
Faner Bldg.
Carbondale, IL 62901

*Warren C. Gibson
Anamet Laboratories, Inc.
Box 831
San Carlos, CA 94070

*Randall Neff
Hewlett-Packard
1501 Page Mill Road
Palo Alto, CA 94306

*Professor Dinh N. Nguyen
Department of Mechanical Engineering
Pavillon Pouliot
Universite Laval
Quebec City, (P. Q.) Canada
G1K 7P4

*Mike Forster
Tektronix, Inc
Box 500
Beaverton, OR 97077

*L. Lopez
3144 Civil Engr. Bldg.
Univ. of Illinois
Urbana, IL 61801

*Richard A. Stahle
Naval Air Station
Code 3451-1
Point Mugu, CA 93042

*Ben McGlamery
Mail Code P003
Visibility Laboratory
University of California, San Diego
La Jolla, CA 92093

*Professor James O. Hossack
University of Nebraska at Omaha
Box 688, DTS
Omaha, Nebraska 68101

*Robert F. Bachert
6570 AMRL/HEB
Wright-Patterson Air Force Base, OH 45433

*G. M. Sigut
%Sulzer Bros. Ltd.
CH-8401 Winterthur
Switzerland
052/814263

*Dr. Werner
Institut F. Bauingenieurwesen I
Der Tu-Muenchen
Arcisstr. 21
8000 Muenchen 2
West Germany

Mike Kersey
Sentcom Technology
Adair Center
3600 Hillcroft, Suite 303
Houston, TX 77000

*Peter Finnigan
General Electric
Materials and Processes Lab
Bldg. 55, Rm. 203
Schenectady, NY 12345

R.W. Linslow
Air Research Manufacturing Co.
P 41
2525 West 190th Street
Torrance, CA 90509

*William E. Johnston
UC Lawrence Berkeley Lab.
One Cyclotron Road
Berkeley, CA 94720

* David T Gregorich
Physics Dept.
California State College
5151 State University Dr.
Los Angeles, CA 90032

Lawrence Doe
Mack Trucks
1999 Pennsylvania Avenue
Hagerstown, Maryland 21740

* Tommy G. Dearmond, Jr.
Department of Computer Science
and Statistics
University of Southern Mississippi
Box 286 Southern Station
Hattiesburg, MS 39401

* Bernard E. Ross,
Center for Mathematical Modeling
University of South Florida
Tampa, FL 33620

* Dr. Hardy Pottinger
and David W. Dearth
Computer Center
Math - CS Bldg.
University of Missouri - Rolla
Rolla, MO 65401

Elliott Dudnik
University of Illinois
Dept. of Architecture
Chicago, IL 60680

* Andre Bouillot
Occidental Petroleum Corp.
Grand Island Data Center
Box 8
Niagara Falls, NY 14302

* Richard E. Roose, Code 1111
US Bureau of Reclamation
Engineering and Research Center
Box 25007, Denver Federal Center
Denver CO 80225

* Norma J. Wilson D632.0
Naval Avionics Facility
21st Street and Arlington Avenue
Indianapolis, IN 46218

* Dr. J. A. Tomas
Department of Mechanical Engineering and
Production Engineering
Royal Melbourne Institute of Technology
124 LA Trobe Street
Box 2476V GPO
Melbourne, Vic. 3000
AUSTRALIA

Dr, G, Dupuis
Civil Engineering Department
Ecole Polytechnique Federale De Lausanne
Eidg Technische Hochschule Lausanne
Politechnico Federale Di Losanna
SWITZERLAND

R. Lasse Henriksen
Ragmilild Szhibbyesv. 36
Oslo 9 NORWAY

* Lynn Seaman
SRI International
333 Ravenswood Ave.
Menlo Park, Calif. 94025

* Tom Sederberg
Dicomed Inc.
9700 Newton Ave. So.
Minneapolis, Minn. 55431

* Ph. Mattenberger
Centre Informatique
EPF-Lausanne
Institut de Technique Des Transports
Chemin Des Delices 9 - 1006 Lausanne
Switzerland