5-1978

Computer Modeling of Cardiac Electrical Activity by Polarcardiogram Patterns

Daniel K. Berry

Abstract

COMPUTER MODELING OF CARDIAC ELECTRICAL ACTIVITY BY POLARCARDIOGRAM PATTERNS

by Daniel K. Berry

The objective of this thesis is to provide for the cardiologist polarcardiograms and additional diagnostic criteria which will enable him to diagnose heart disease more easily. Other contributions are those of: (a) providing a system of data acquisition and (b) analyzing Dr. Gordon Dower's criteria for myocardial infarction using the programs provided in this thesis.

The problem inherent in the reading of a polarcardiogram is the identification of the "loop" patterns with the associated heart disease. This can be accomplished by reading and interpreting both vectorcardiograms and electrocardiograms as well as angiograms, then relating the interpretation to the polarcardiogram patterns.

The problem of computer programming and transfer of data from the Dower cart to the MIIS system to the Tektronix terminal, then plotting the "loops" requires the following skills in completion of the task: expertise in computer operation, knowledge and understanding of cardiac electrical activity and biomath computation.

An additional problem encountered in the immediate completion of the objectives was due to the fact that the Roche system, which converts a signal from an analog wave to a digital form, was not yet installed at the beginning of the project. The Roche system is a computerized

system for EKG interpretation and analysis. However, it was possible to substitute the EMR 6130 Fortran system and the PDP 12, which although less efficient, accomplishes the same results as the Roche system to obtained digitized data.

The three processes beginning with the EMR 6130 Fortran, converting to the PDP 12 and EMR 6130 are as follows:

1. Frank-lead electrocardiogram data were collected on the Dower cart, processed on the EMR 6130, then plotted on the Calcomp plotter.
2. Frank-lead electrocardiogram data were collected on the Dower cart, transferred to the EMR 6130, stored in the MIIS operating system, processed in the Tektronix terminal and then plotted on the Tektronix plotter.
3. Frank-lead electrocardiogram data were collected on the Dower cart, digitized by the PDP 12, transferred to the EMR 6130 for pattern recognition, stored in the MIIS operating system, processed in the Tektronix terminal, and then plotted on the Tektronix plotter.

Procedures followed to complete this study were:

1. Collect the Frank-lead electrocardiogram data on the patients.
2. Convert the analog wave pattern to digitized numerical data.
3. Do the necessary pattern recognition to produce a median wave.
4. Store the median wave as numerical data.
5. Run the vectorcardiogram program to plot the vectorcardiograms.
6. Run the 12-lead synthesis program which synthesizes the 12 scalar leads from the 3 Frank leads.
7. Run the polarcardiogram program to produce the polar patterns on

an Aitoff projection.

8. Run the transverse magnitude vs. beta wave and sagittal magnitude vs. gamma wave programs.

9. Diagnose disease by use of the angiogram used as the gold standard.

10. Relate each disease to the transverse magnitude vs. beta wave sagittal magnitude vs. gamma wave.

11. Correlate the results of the magnitude vs. beta and gamma waves with Dr. Dower's criteria for myocardial infarction.

Summary

The following myocardial infarctions have been identified by the Polar-cardiographic criteria of the transverse magnitude vs. beta wave and sagittal magnitude vs. gamma wave: (Dr. Dower's Criteria)

(a) Anterior myocardial infarction:

1. TM_{QRS} returns to zero after an initial deflection.
2. $Beta_{QRS}$ has an upward slope before TM_{QRS} reaches its maximum.
3. $Beta_{QRS}$ is entirely positive.

(b) Inferior myocardial infarction:

1. SM_{QRS} returns to zero after an initial deflection.
2. $Gamma_{QRS}$ has a downward slope before SM_{QRS} reaches its maximum.

Conclusion:

Based on the results of this project on 20 patients it is possible to use the criteria for myocardial infarction from the magnitude vs. angle waves for the following:

1. Diagnosis of Anterior myocardial infarction.
2. Diagnosis of Inferior myocardial infarction.

LOMA LINDA UNIVERSITY

Graduate School

COMPUTER MODELING OF CARDIAC ELECTRICAL ACTIVITY BY POLARCARDIOGRAM PATTERNS

by

Daniel K. Berry

A Thesis in Partial Fulfillment

of the Requirements of the Degree, Master of Science

in the field of Biomathematics

May 1978

Each person whose signature appears below certifies that this thesis in his opinion is adequate, in scope and quality, as a thesis for the degree Masters of Science.

Ivan R. Neilsen, Chairman
Ivan R. Neilsen, Chairman of Biomathematics

C. Duane Zimmerman
Duane Zimmerman, Professor of Biomathematics

Roy Jutzy
Roy Jutzy, Cardiologist

TABLE OF CONTENTS

	Page
LIST OF FIGURES	iv
Chapter	
1. CARDIAC ELECTRICAL ACTIVITY	1
CARDIAC PHYSIOLOGY	
GENERATION OF ELECTRICAL ACTIVITY	
HISTORY OF ELECTROCARDIOGRAPHY	
2. COORDINATE SYSTEM	4
FRANK LEADS	
X, Y, Z COORDINATES	
3. PRODUCTION OF A MEDIAN WAVE	6
4. VECTORCARDIOGRAMS	9
5. TWELVE-LEAD SCALAR SYNTHESIS	21
6. POLARCARDIOGRAMS	38
7. DIAGNOSTIC CRITERIA BY MAGNITUDE VS. ANGLE WAVES	48
BIBLIOGRAPHY	54
APPENDIXES	
1. COMPUTER PROGRAMS AND DATA	57
2. VECTORCARDIOGRAM PROGRAMS AND SAMPLES	77
3. TWELVE-LEAD SCALAR SYNTHESIS PROGRAMS AND SAMPLES	90
4. POLARCARDIOGRAM PROGRAMS AND SAMPLES	106

LIST OF FIGURES

Figure		Page
1.	The Heart Vector in the Center of a Sphere	2
2.	Electrocardiogram Wave Pattern Affected by Undulating Breath Activity	8
3.	Electrocardiogram Wave Pattern with the Base Line Tied Down	8
4.	Median Wave for One Cardiac Cycle	8
5.	X, Y, and Z Waves	10
6.	Vectorloops	12
7.	Vectorloops with P and T Wave Moved	13
8.	Vectorloops (A) produced by the EMR 6130	14-16
	(B) produced by the Tektronix terminal	17-19
9.	Horizontal Plane Transparency	23
10.	Transversal Plane Vectorcardiogram	24
11.	Frontal Plane Transparency	25
12.	Frontal Plane Vectorcardiogram	26
13.	Normal Axis	28
14.	Transformed Axis	28
15.	Transformed Axis Superimposed on the Normal Axis	28
16.	Point Located on a Normal Axis	28
17.	Point Located on a Transformed Axis	28
18.	Illustration for transformation of coordinate systems	29
19.	Twelve scalar leads synthesized by a program in Fortran running on the EMR 6130	32-33

20. Twelleads synthesized by a program in Basic on the Tektinal 34-37

21. Aitoion 39

22. Coordsformation 42

23. Polarline 44

24. Polaritoff projection produced by the EMR 6130 46

25. Polaritoff projection produced by the Tektronix termi: 47

26. Patie 49

27. Polar criteria for infarction 50

CHAPTER 1

CARDIAC ELECTRICAL ACTIVITY

The heart is surrounded by body fluids which contain a large amount of electrolytes. This fluid is both inside and outside of body cells. The main electrolyte is sodium chloride--common table salt--which has about the same concentration as sea water. The main electrolyte inside the cell is potassium. These electrolytes make the body a good electrical conductor. Thus any electrical activity in the body produces an electrical field throughout the entire body.[1] Action potentials which generate muscle activity causes an increase in muscle permeability which permits the sodium to flow into the muscle cells and potassium to flow out of the cells.

The transfer of these charged particles (ions) to the opposite side of the cell walls generates an electrical field. This electrical field can then be detected by electrodes placed on the surface of the body. The heart muscle is the largest generator of electrical fields. Other muscles can also generate electrical fields, and therefore it is best to take an electrocardiogram on a resting patient.

The electrical activity of the heart starts at the sino-atrial node, --the "pacemaker"-- and then moves throughout the rest of the heart. Electrodes may be placed on the surface of the body, which can detect the strength and direction of the electrical impulse at any moment in time. The strength and direction of the electrical impulse can be interpreted as a vector with magnitude and direction. As time progresses throughout the cardiac cycle the head of the vector forms a loop which is three-dimensional.

William Einthoven, a Dutch scientist, originated the theoretical concept of the electrocardiogram. He considered the heart acted like an electrical dipole with a positive and a negative end in the center of a sphere of saline solution. (He perceived the body as a sphere.)[2] Therefore an electrode placed on the surface of the sphere could detect the magnitude of the electrical activity coming straight toward that point on the surface where the electrode was positioned.

Therefore, if a second electrode were placed at another point on the surface of the sphere, the magnitude of the electrical impulse in that direction could also be detected.

With these two electrodes (leads) a vector could be plotted as shown in Figure 1.

The heart is in the center
of the sphere.
The vector is indicated
by the arrow.

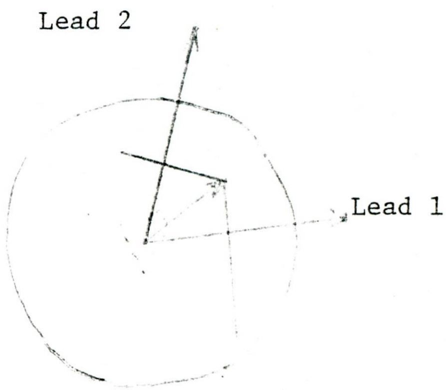


Figure 1

Although Einthoven first discovered electrocardiology in 1902, it did not become popular for use until 1949 when technological development made it practical to use. A vectorcardiogram is a two-dimensional graphic picture of the heart's electrical loop as it would appear from any given plane. The usual planes that are used for the vectorcardiograms are the frontal, transversal and sagittal planes, (front, top, and side). Vectorcardiography first began in 1936 and is just now becoming useful for routine clinical use.

A polarcardiogram is a graphic picture that portrays the heart electrical loop as a three-dimensional "polar loop" or polar pattern. Since the polar loop is graphed on a two-dimensional paper, a method can be used which causes the loop to appear to be three-dimensional.

Polarcardiography began in 1937 but was not commercially available until 1968. So far, it has been used mainly for research and is not yet regularly used for clinical diagnosis.[3]

CHAPTER 2

COORDINATE SYSTEM

The Frank-lead system is a positioning of electrodes on the body in such a manner as to detect the electrical activity of the heart from the equivalent of a side, top, and front view of a patient. These views in exact sequence of side, top, and front, reveal the electrical activity corresponding to X, Y, and Z coordinate system.

The Frank leads are not actually placed on the side, top, and front of the patient's body, because it is not a perfect sphere nor is the entire body all of the same density. Furthermore, not all of the body tissue conducts the electrical current equally well. Therefore the Frank leads are positioned in such a manner as to gather data as if the leads were positioned on a perfect sphere of the same density and electrical conductivity, so as to produce an X, Y, Z lead system.

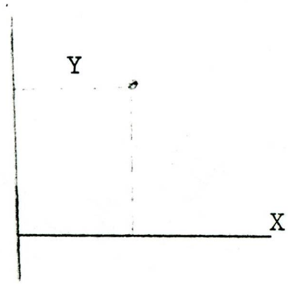
The Frank leads are positioned as follows:--on the back, the neck, front chest, right chest, right leg, left chest and left nipple.

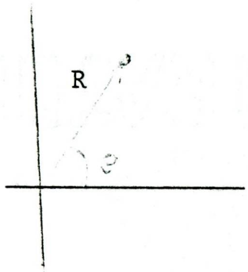
Since the X, Y and Z coordinates are found from the Frank lead electrocardiogram, any point in space can be located by a set of three numbers corresponding to the X, Y and Z. Therefore, any point in space can also be located by spherical coordinates by a simple mathematical conversion.

From this information, the twelve-lead electrocardiogram can be synthesized from the three Frank leads. Also this establishes the basis for which vectorcardiograms and polarcardiograms can be produced by computers.

Shown below are the equations for the conversion from polar to rectangular coordinates.

$$X=R \cos 0$$
$$Y=R \sin 0$$





CHAPTER 3

PRODUCTION OF A MEDIAN WAVE

A "median wave" is the pattern that is produced by obtaining the median of all the waves taken over a period of ten seconds. It is necessary to show how a median wave is established for one cardiac cycle for a patient.

First, the data from the Frank leads are collected from the patient on an electrocardiogram cart. Then these X, Y, Z Frank lead patterns are run through the Roche system computer where they are digitized. Each of the three are sampled 250 times per second for ten seconds. This data is then transferred to the MIIS system where it is stored. (See Appendix 1.1)

These data are then transferred to the Tektronix terminal. (See Appendix 1.2, 1.3) The Tektronix terminal has been programmed in Basic so that it recognizes wave patterns and produces a median wave for one cardiac cycle. The program is based on the following facts:

1. An undulating breath pattern alters the base line of the cardiac cycle. (See Fig. 2). Therefore the base line must be "clamped down" or "tied down" so that the base line is at zero. (See Fig. 3). This is done by recursively going over the ten seconds of the data and establishing the zero position, then adjusting the wave so that it appears as in Figure 3. The wave pattern is identified by finding the slope and peak of the R wave.

2. Once the cardiac cycle has been established, an average of all the cycles is found by taking the median point at each position of the wave that was sampled.

3. The median provides a better wave than the mean since the median is not as seriously affected by spikes or other large or small irregularities in the wave pattern as is the mean. The final result is a digitized form of the median wave of one cardiac cycle with a zero base line as illustrated in Figure 4. This is done for the X, Y, and Z leads. (See Appendix 1, Examples 4, for the Median Wave Program in Fortran and Example 5 for the Median Wave program in Basic on the Tektronix). (See Appendix 1, Example 6 and 7 for the plotted Median Waves.)

The "undulating breath" pattern.

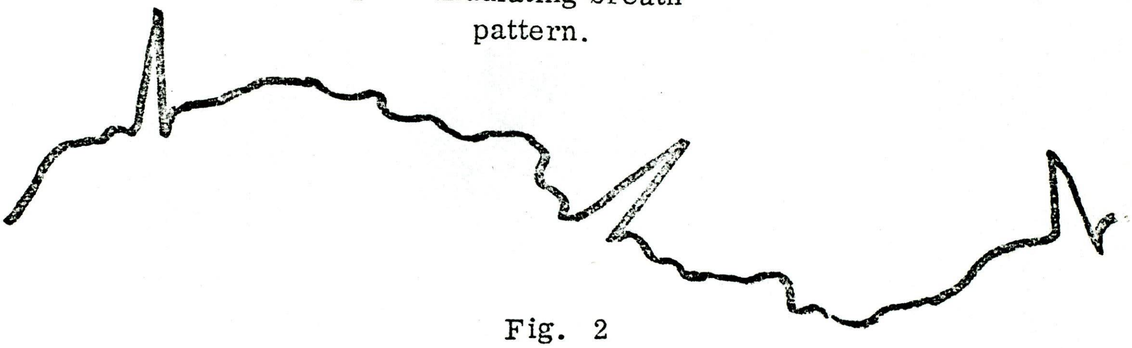


Fig. 2

Base line "clamped down"

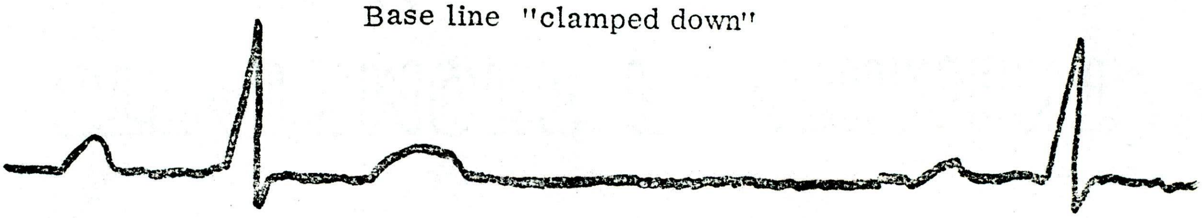


Fig. 3

A Median Wave.

Fig. 4

CHAPTER 4

VECTORCARDIOGRAMS

Vectorcardiograms are produced by the use of the X, Y, and Z median waves of one cardiac cycle. The median wave data is a set of ordered pairs, such as (X,t) where X is the displacement of the cardiac electrical activity in the X or side direction and t is time in 250th's of a second. One complete cardiac cycle is approximately .7 of a second. (Y,t) and (Z,t) are defined similarly.

The data are then plotted by the first element of the ordered pair vs. the second element of the ordered pair. Then each point is connected to the next point by a line. The result is a plotting of the cardiac electrical activity for one cycle in the X, Y, or Z direction, which is then the pattern produced by the X, Y, or Z wave. This pattern appears similar to the patterns produced by the twelve leads obtained from the electrocardiogram cart. The X, Y, Z patterns are simply another view of cardiac electrical activity. The result can be seen in Figure 5 on the following page. This pattern was produced by a Fortran program on the EMR 6130 computer. The parts of the wave are labeled P, QRS, and T, which are the characteristic parts of an electrocardiogram.

Vectorcardiograms are also sets of ordered pairs. The three sets of ordered pairs are (X,-Y), (X,Z) and (Z,-Y). These data are also plotted by the first element of the ordered pairs versus the second element of ordered pairs for one entire cardiac cycle, just as the three waves,

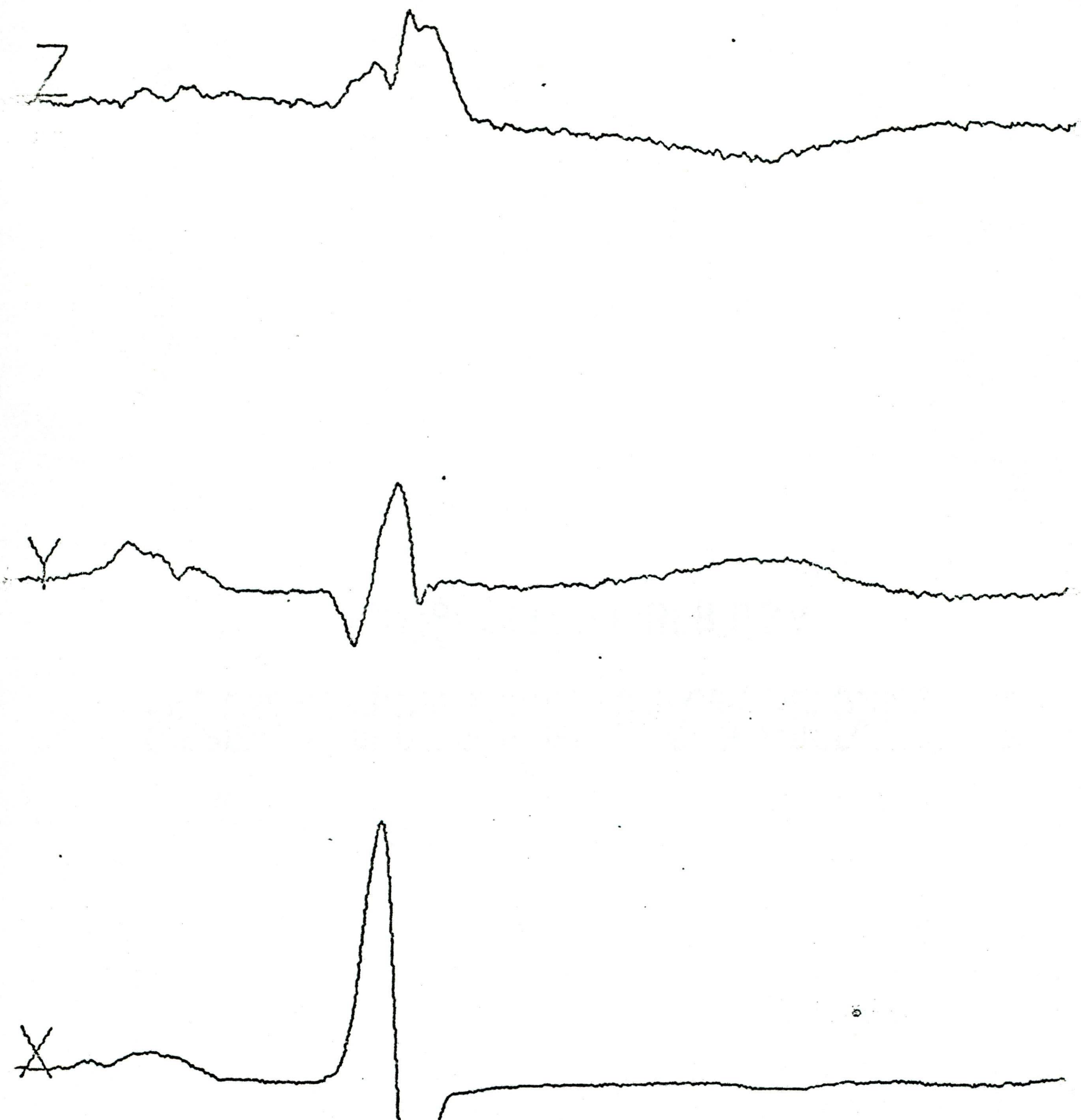


Fig. 5

(X, Y, and Z) previously explained. The three plottings of (X,-Y), (X,Z) and (Z,-Y) produce three loops. (See Fig. 6.)

Similar to an electrocardiogram the vectorcardiogram consists of three loops--the P loop, QRS loop, and the T loop, which are produced by the P wave, the QRS wave and the T wave.

The vectorloop is a projection of the cardiac electrical activity onto a plane. Therefore, the three dimensional electrical activity is a view from three different positions: front, top and side on two dimensional planes. These are called the frontal, transversal and sagittal planes.

In order that the three waves can be seen clearly it is necessary to move them apart so that they will not be on top of each other. The P wave is moved to the left and the T wave is moved to the right. The QRS wave is left in its original position. (See Fig. 7.)

The cardiologist needs to know how fast the vectorloop is being produced. Therefore, timing marks were placed at every 500th of a second, two milliseconds. Also the cardiologist needs to know which direction the loop is going. To reveal this an arrow is placed on the front of a timing mark or a dot is placed on the tail of a timing mark. (See Fig. 8A and Fig. 8B). Figure 8A was produced by a Fortran program on the EMR 6130, and Figure 8B was produced by a Basic program on the Tektronix terminal.

See Appendix 2, Example 1 for the Fortran program for the vectorcardiograms.

See Example 2 for the vector loops produced by the Fortran program.

See Example 3 for the Tektronix Basic program for vectorcardiography.

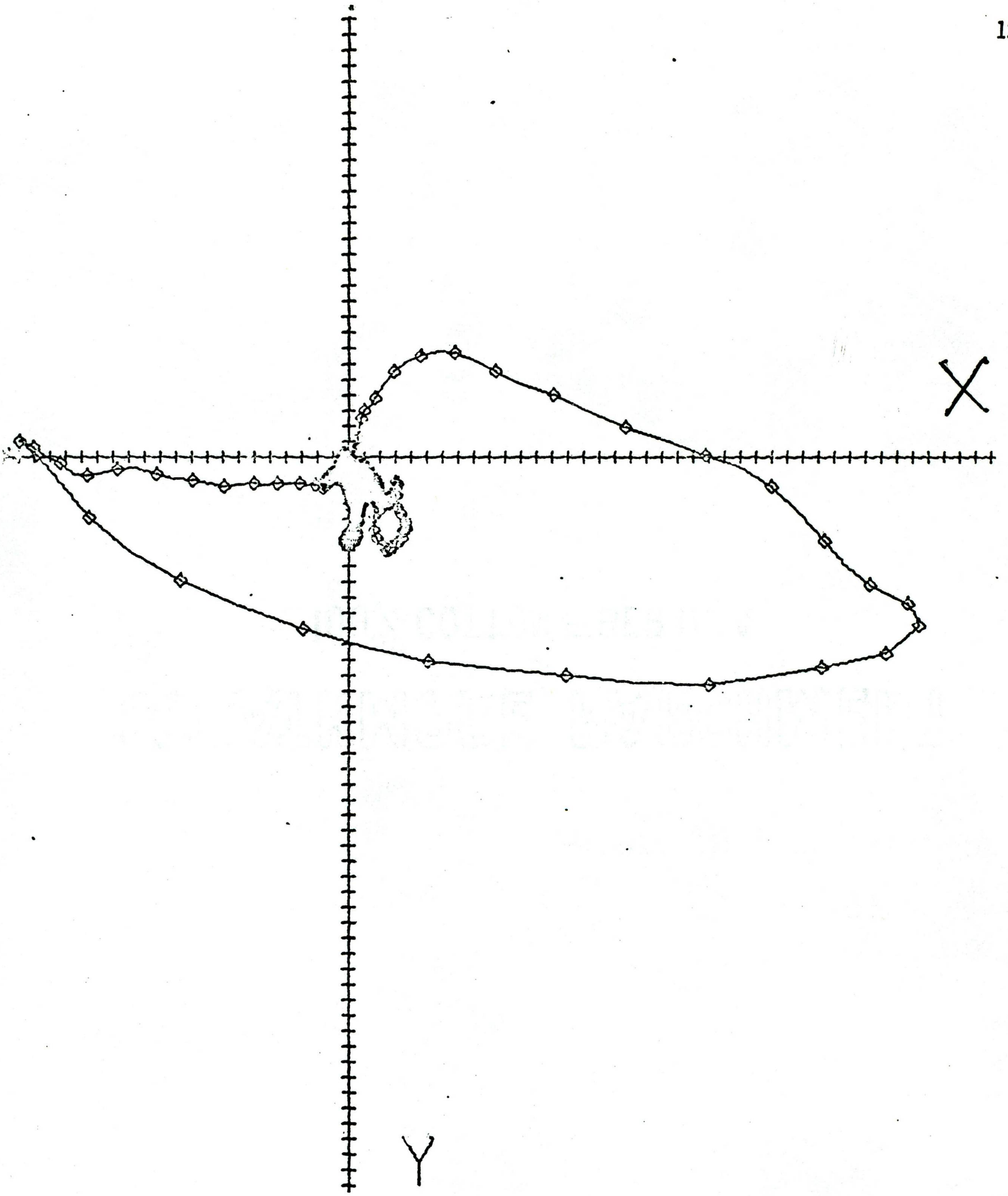


Fig. 6

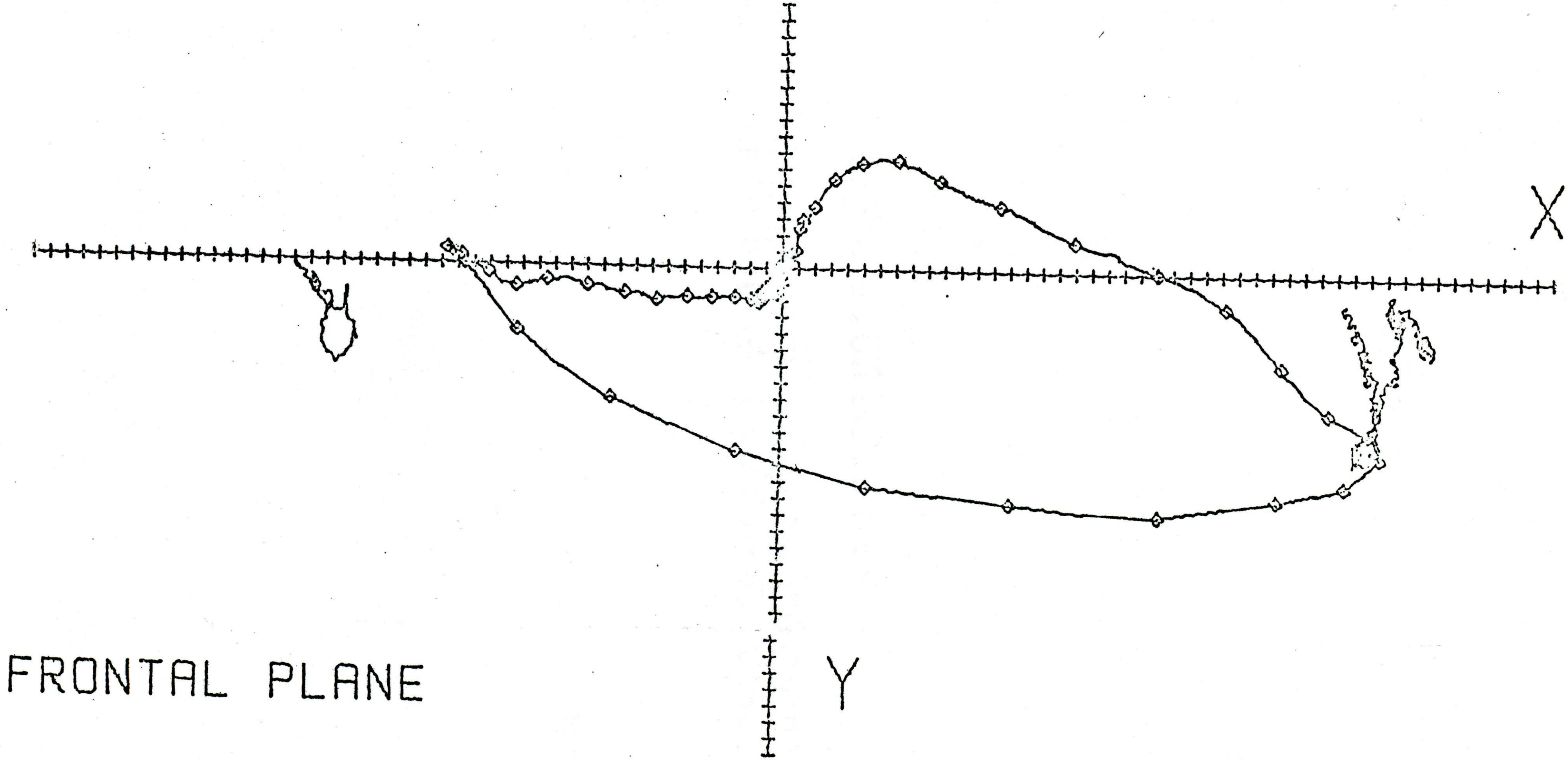


Fig. 7

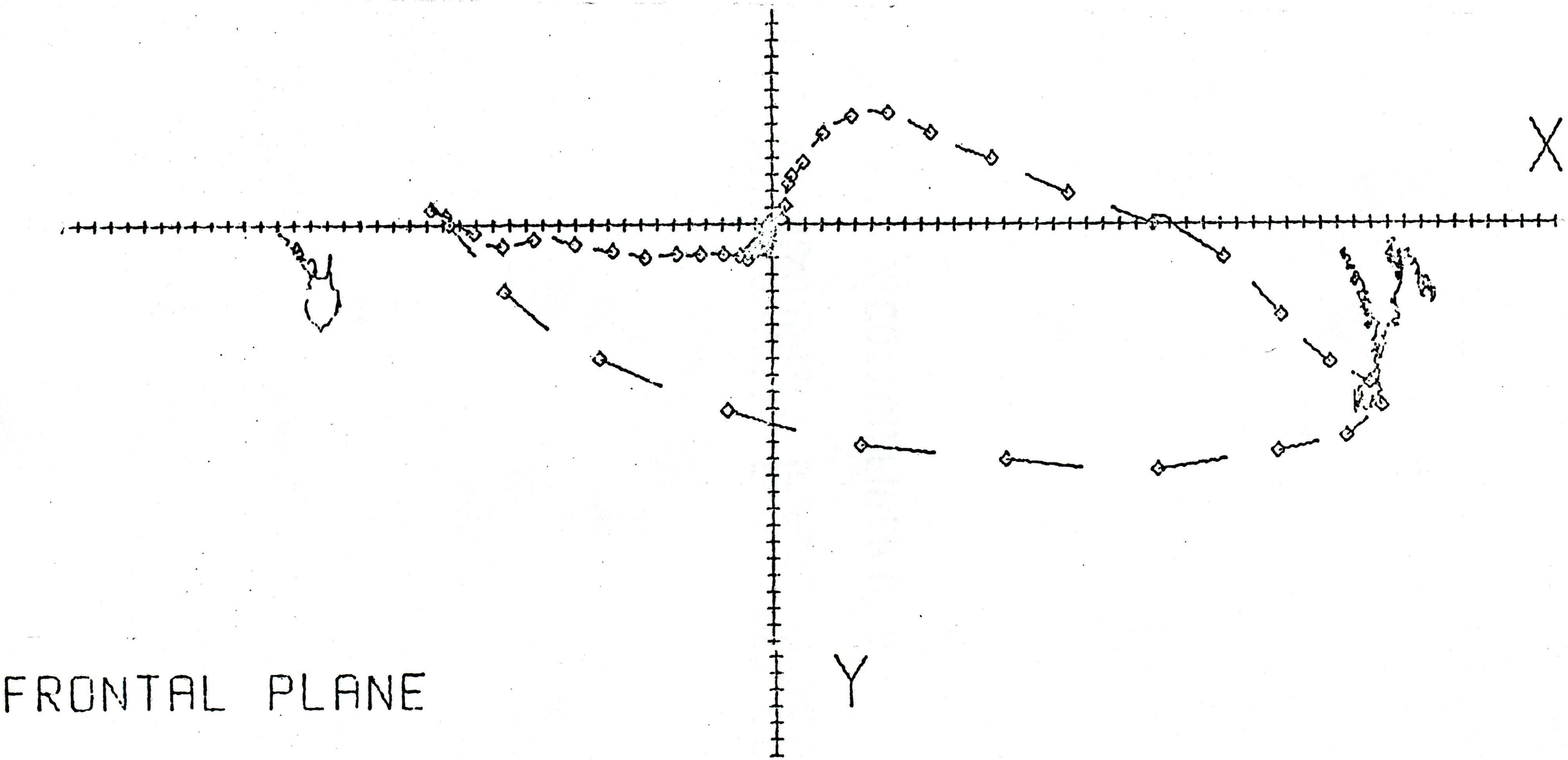
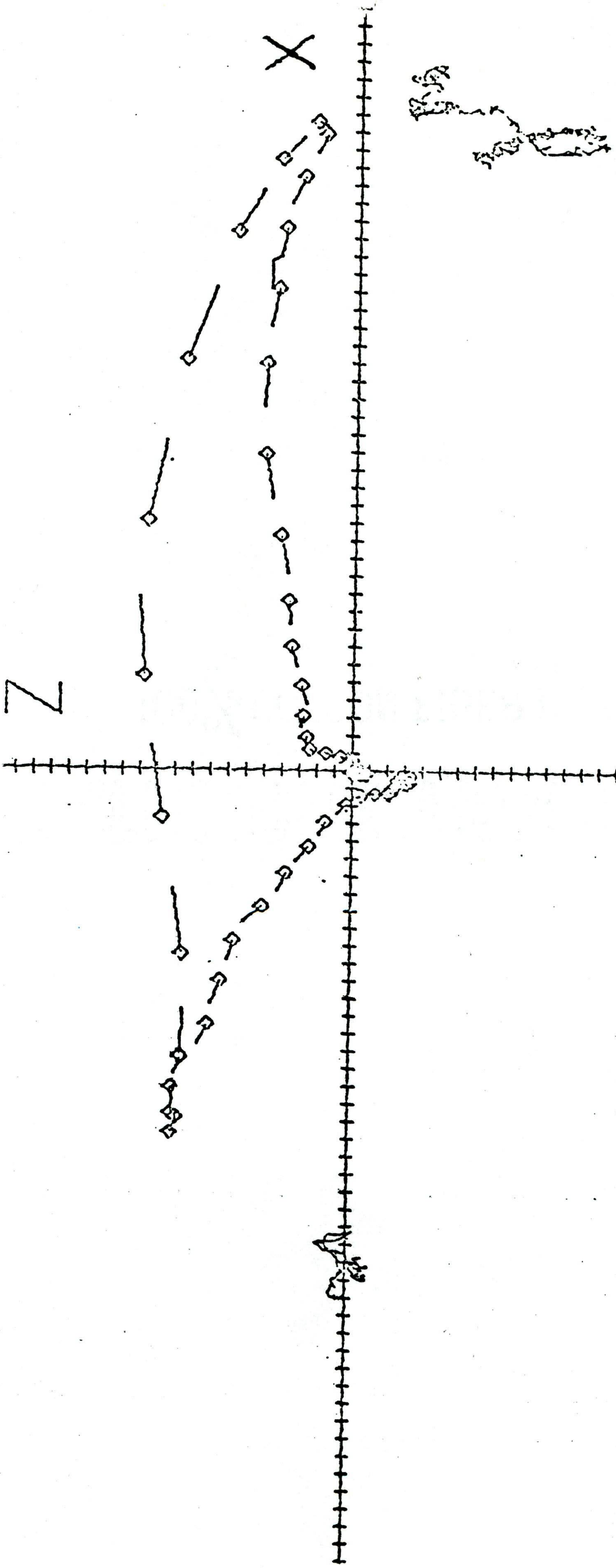


Fig. 2A



TRANSVERSE PLANE

Fig. 8A

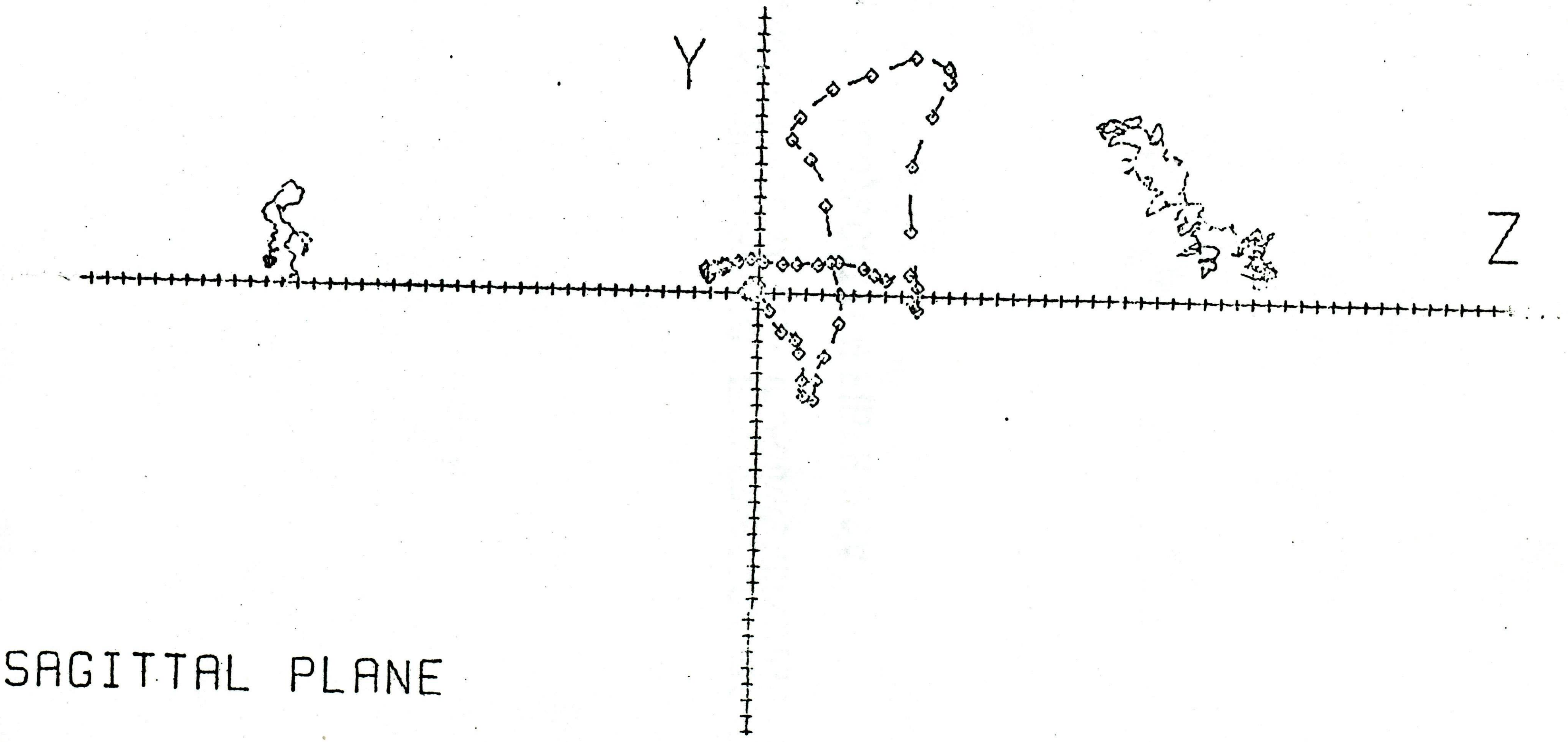


Fig. 8A

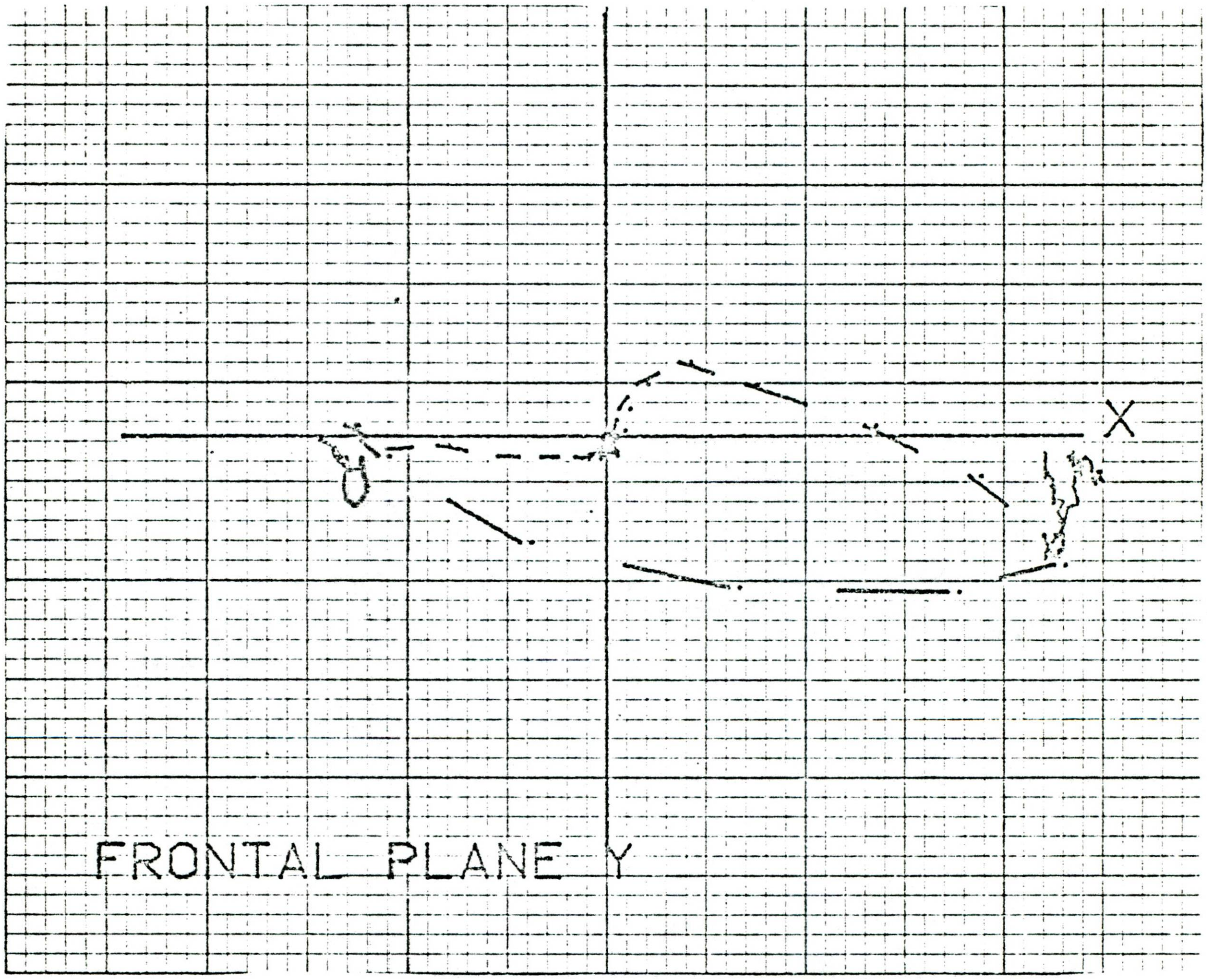


Fig. 8B

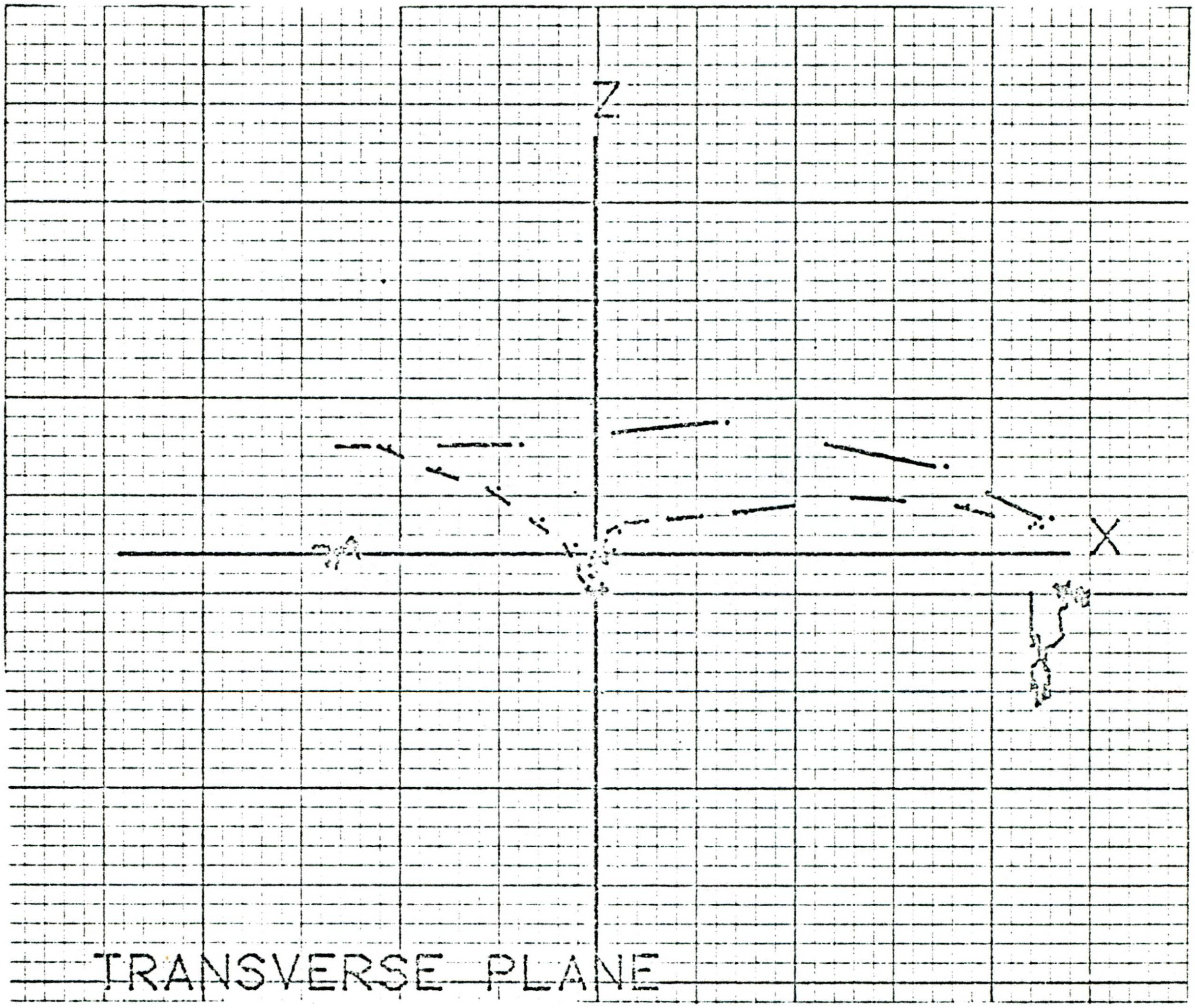


Fig. 8B

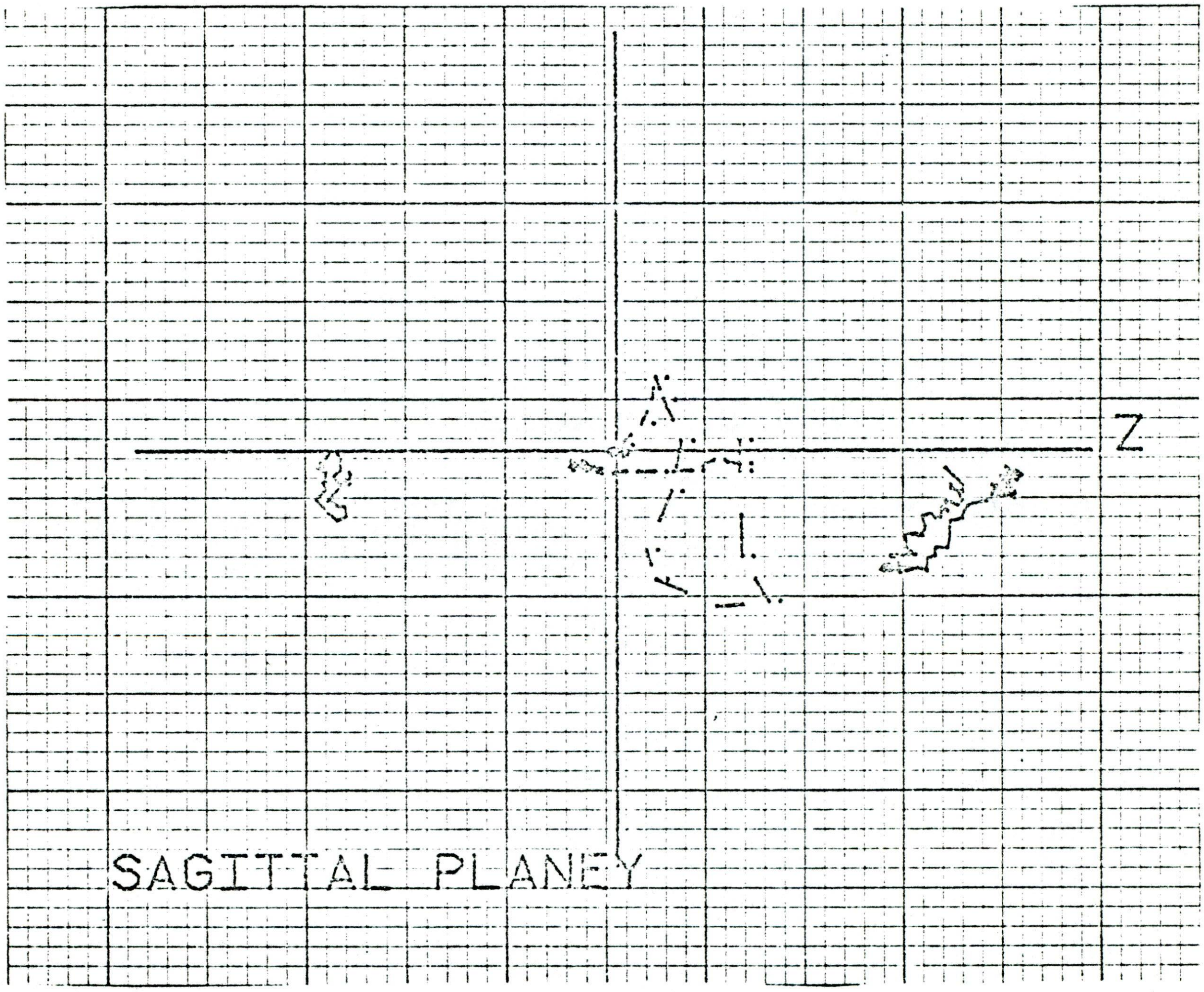


Fig. 8B

See Example 4 for the Vector loops produced by the Basic program on the Tektronix terminal.

CHAPTER 5

SYNTHESIS OF TWELVE SCALAR LEADS

The twelve scalar leads are twelve different views of the cardiac electrical activity. There are six chest leads which view the electrical current of the heart as either coming toward or going away from that particular point on the chest, and also the magnitude with which the electrical current is moving. The other six leads view the electrical activity in a manner similar to that of the chest, but from the arms and legs.

An advantage of synthesizing the twelve scalar leads from the three Frank leads is that when a vectorcardiogram or polarcardiogram is taken, the Frank leads are already connected to the patient. Therefore no new set of leads or placement of electrodes must be taken in order to obtain the twelve scalar leads. Another advantage of the synthesis of the twelve scalar leads from the three Frank leads is that the Frank leads require fewer electrodes than do the twelve scalar leads and thus save effort and time.

Since the Frank leads provide an X, Y and Z coordinate for each point of the digitized activity, any point is easily located in space. Therefore, viewing the electrical activity from different positions would require transformations of coordinate systems. Since the twelve scalar leads are different views of the electrical activity, they would require transformations of coordinate systems.

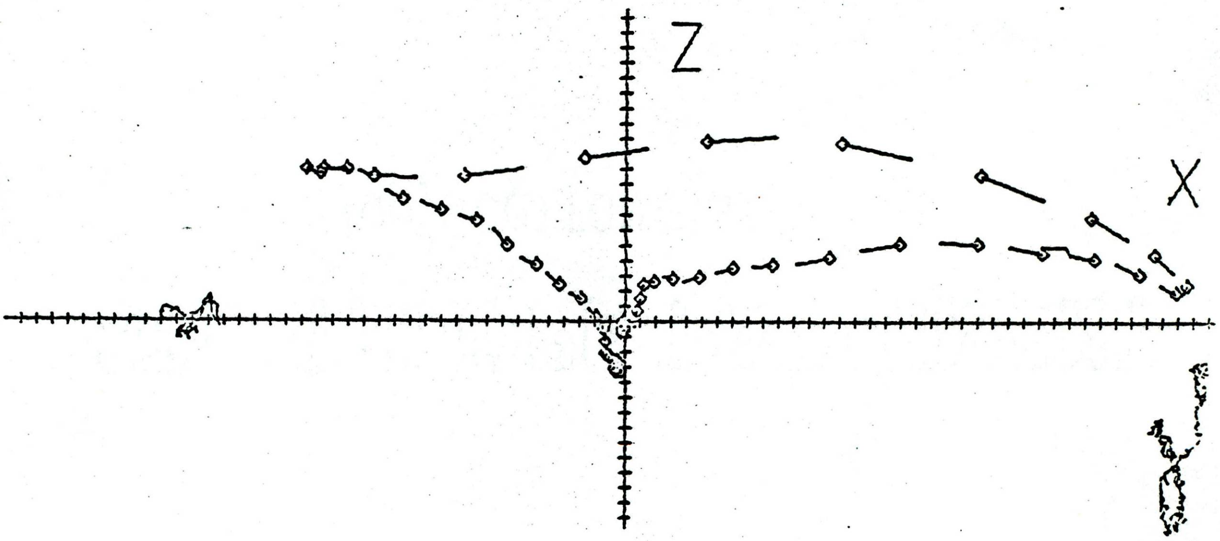
Before explaining the mathematical equations which accomplish this transformation of coordinate systems, it is necessary to explain the visual method of synthesizing the twelve leads from the vectorcardiograms produced by the Frank leads.

The six chest leads lie in the transverse plane (horizontal plane). Therefore, the six chest leads can be synthesized by viewing the vectorcardiogram in the transverse plane from six angles. The six angles are determined by the position of the electrodes on the chest. The other six leads can be synthesized similarly by use of the vectorcardiogram in the frontal plane. (See Fig. 9, 10, 11, 12.)

Figures 9 and 11 are transparencies with the appropriate angles drawn in. The transparencies are then placed over their respective planes, with the zero point of the transparency directly over the zero point of the plane. The lines on the transparencies are labeled with the lead that corresponds to the angle of the line.

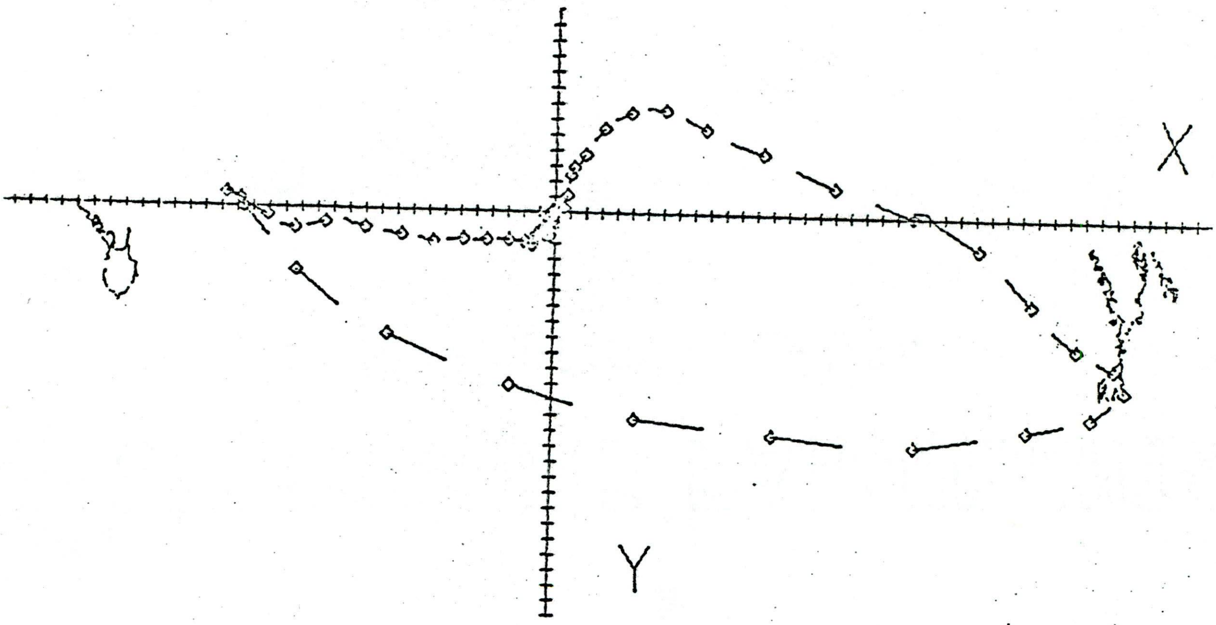
In order to synthesize a particular lead, the following method is used:

1. Choose the lead to be synthesized.
2. Find the line on the transparency labeled with the lead.
3. Find the line perpendicular to the line labeled with the lead, that passes through zero.
4. Define Y_1 to be the shortest distance from any point on the vectorcardiogram to the perpendicular line.
5. Let the Y_1 be negative if the point is on the opposite side



TRANSVERSE PLANE

Fig. 10



FRONTAL PLANE

Fig. 12

of the perpendicular line from the label. Otherwise Y_1 is negative.

6. Find Y_1 for every point on the vectorcardiogram.
7. Arrange the Y_1's in a chronological sequence.
8. Let the sequence be labeled Y_{1i} for i=1 to N, where N is the total number of points in the vectorloop.
9. Let t_i =i for i=1 to N.
10. Let (Y_{1i}, t_i) for i to N, be a set of ordered pairs.
11. Graph the set of points (Y_{1i}, t_i) for i=1 to N.
12. Connect the i^{th} point to the $i+1^{th}$ point for t=1 to N-1.

The mathematical method for synthesizing the twelve scalar leads is the same as the above except for step 4.

Y_1 is defined in the following section:

A. Define the vertical line to be the Y axis.

B. Define the horizontal line to be the X axis. (See Fig. 13)

C. Define the lead line to be the Y_A axis.

D. Define the perpendicular line to be the X_A axis. (See Fig. 14)

E. Let M be any point on the vectorcardiogram.

Then, M can be located by giving an X and a Y coordinate, as in Figure 16, or by an X_1 and a Y_1 coordinate as in Figure 17.

X and Y are known since they come from the Frank-lead digitized data.

But X_1 and Y_1 are not known. It is necessary to find Y_1 in order to synthesize the twelve scalar leads as was outlined in the preceding

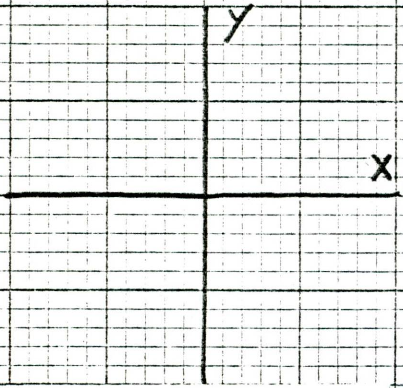


Fig. 13

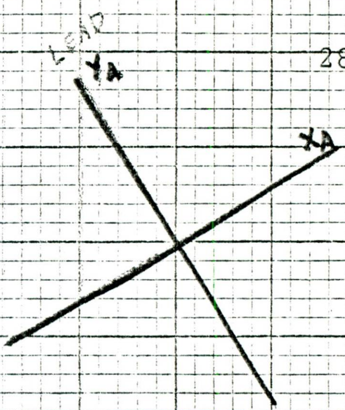


Fig. 14

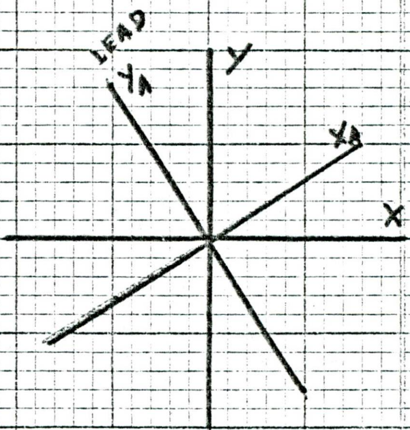


Fig. 15

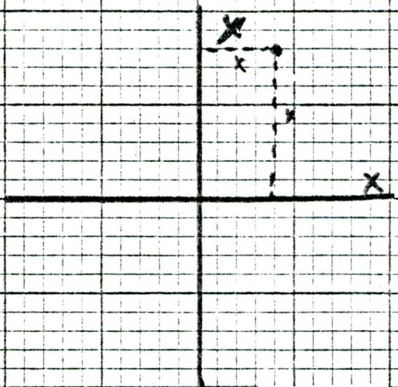


Fig. 16

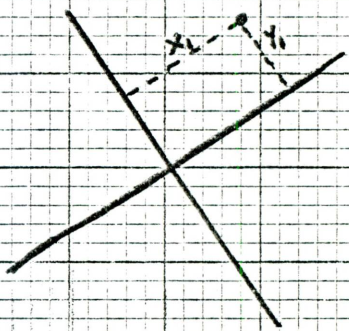


Fig. 17

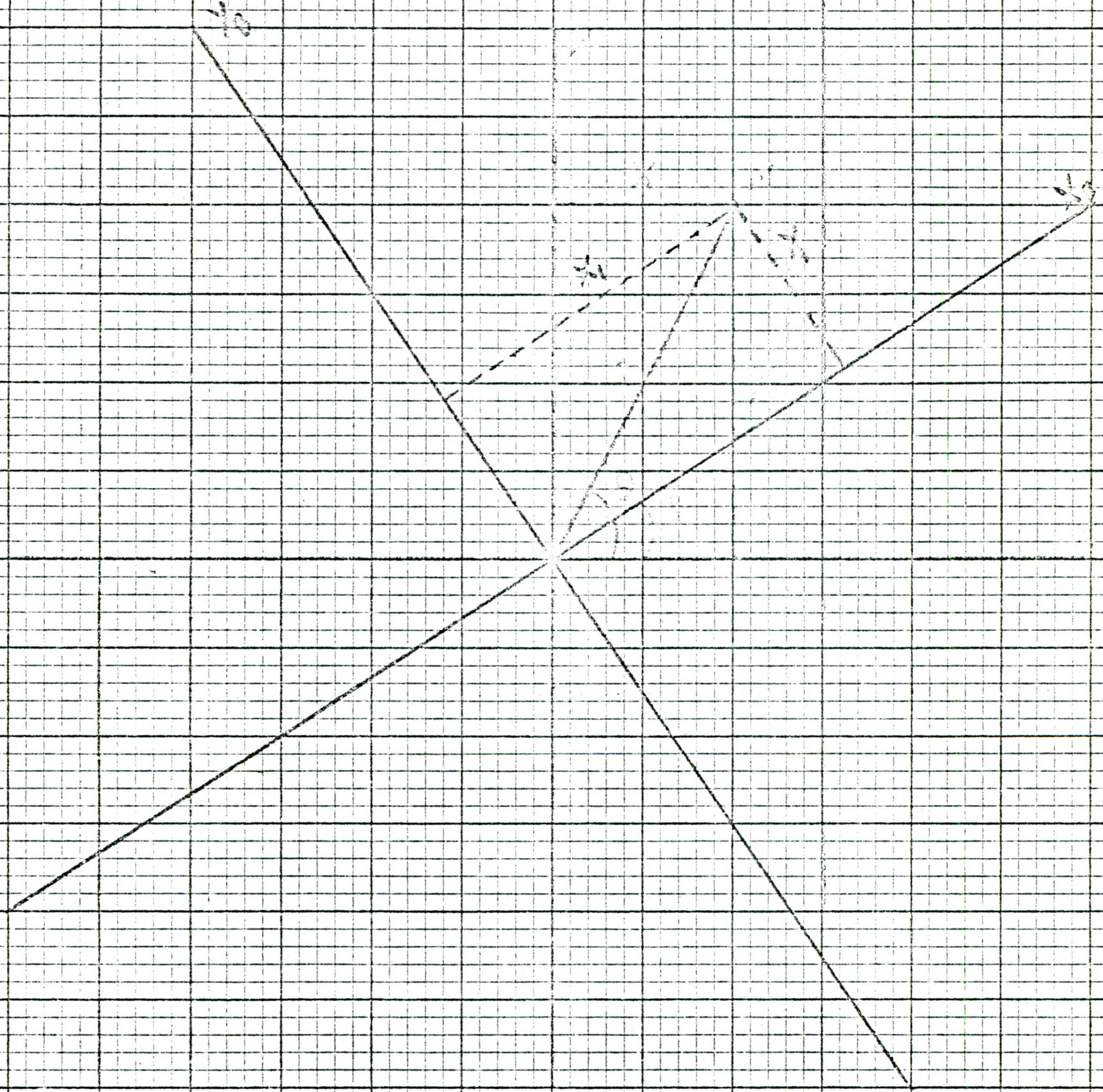


Fig. 18

methods (steps 1-12). Therefore, Y_1 must be found in terms of X and Y, as shown in the equation below:

$$Y_1 = (\sqrt{X^2 + Y^2}) . \sin (\tan^{-1} (\frac{Y}{X}) - Q)$$

Q is the angle of the lead line.

This equation was derived from the following three equations:--the following three equations are derived from Figure 18. (See Fig. 18).

1.) $R = \sqrt{X^2 + Y^2}$

2.) $\sin (\theta - Q) = \frac{Y_1}{R} == Y_1 = R \sin (\theta - Q)$

3.) $\tan \theta = \frac{Y}{X} == \theta = \tan^{-1} (\frac{Y}{X})$

Thus by subtraction

$$Y_1 = (\sqrt{X^2 + Y^2}) \sin (\tan^{-1} (\frac{Y}{X}) - Q)$$

The previous method for synthesizing the twelve scalar leads is used with Y_1 defined by the preceding equation.

(See Fig. 19 for the twelve scalar lead synthesis by a Fortran program on the 6130.)

(See Fig. 20 for the twelve scalar lead synthesis by a Basic program on the Tektronix terminal.)

See Appendix 3, Examples 1 and 3 for the Fortran and Basic programs that synthesize the twelve scalar leads.

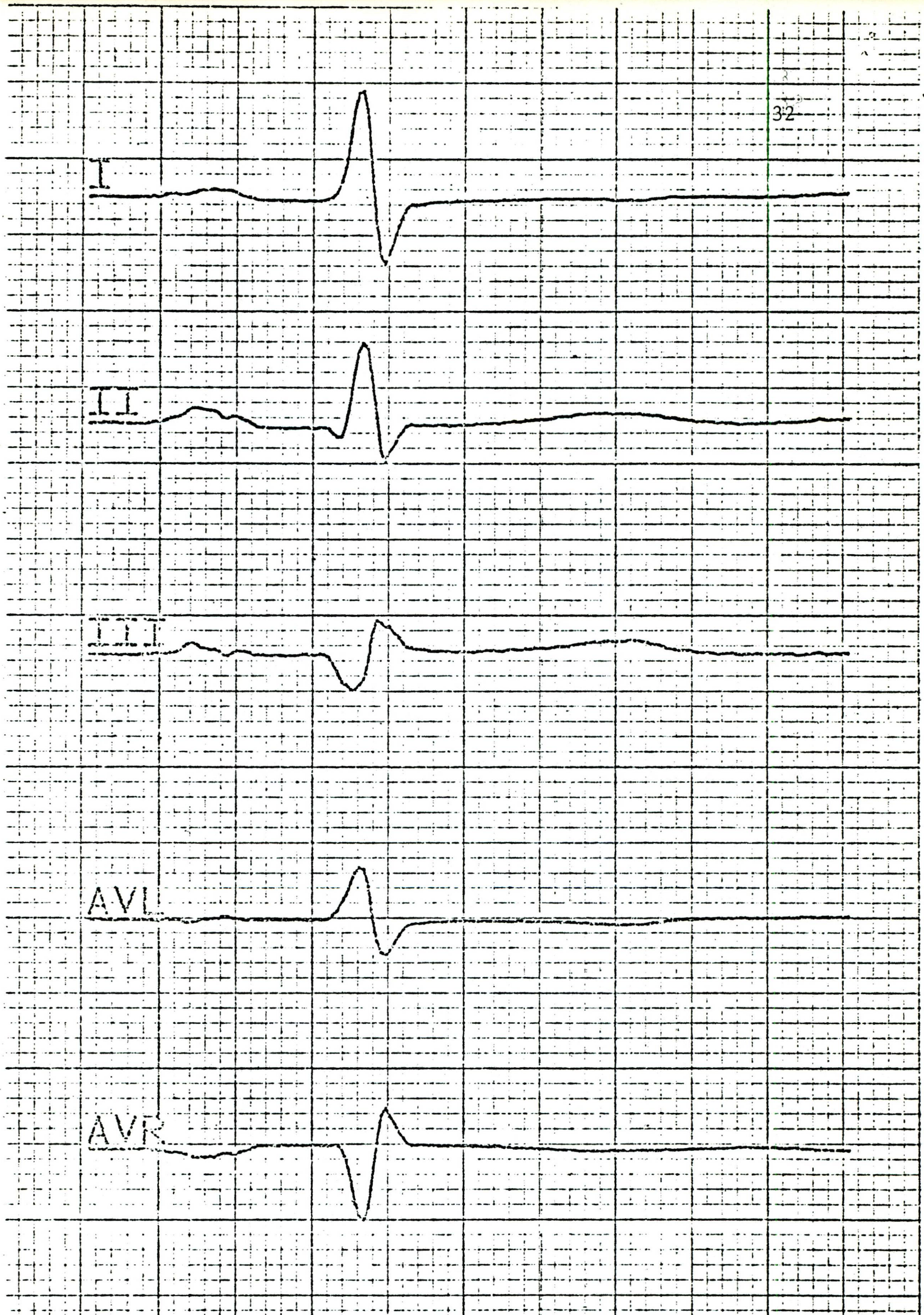


Fig. 19

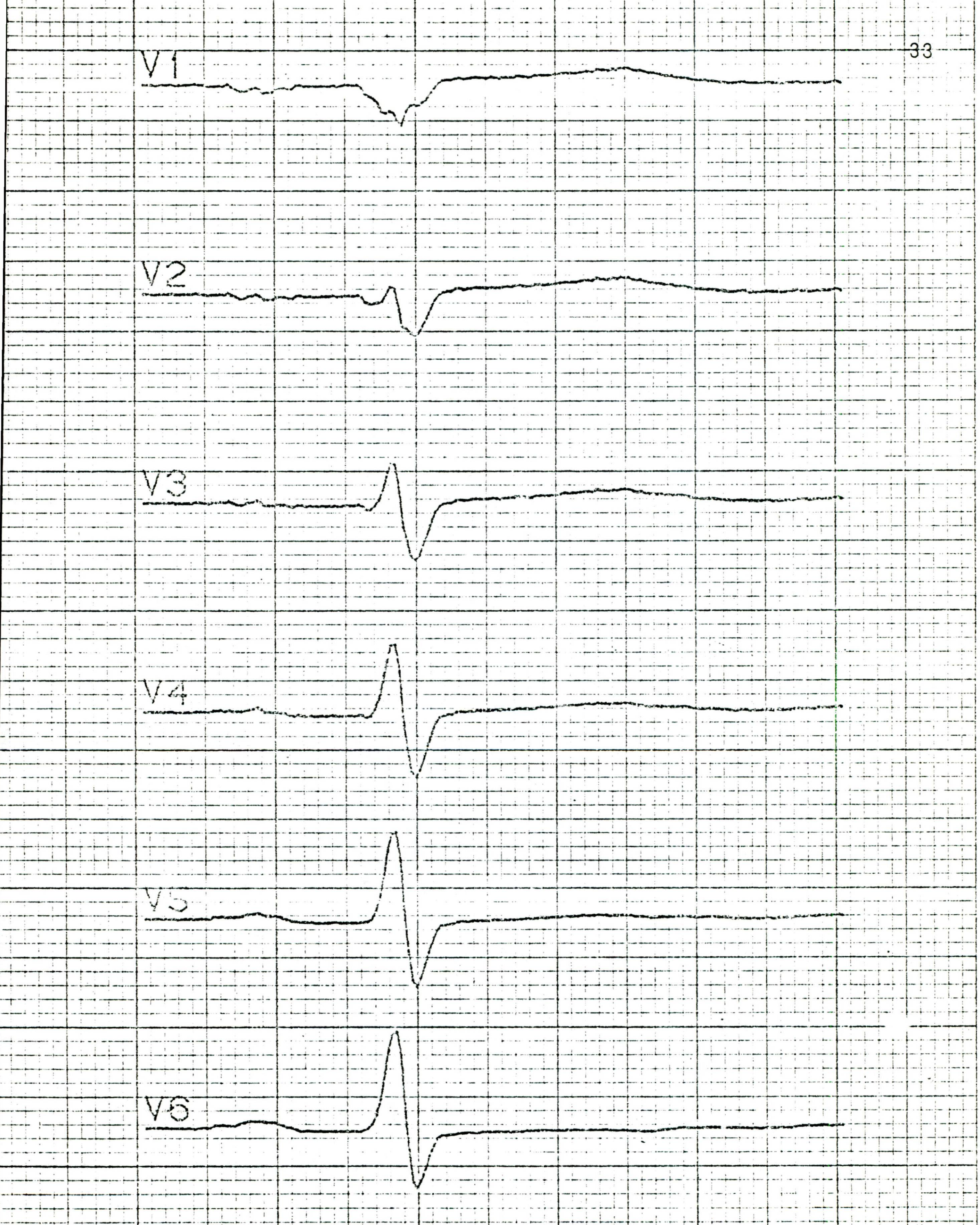


Fig. 19

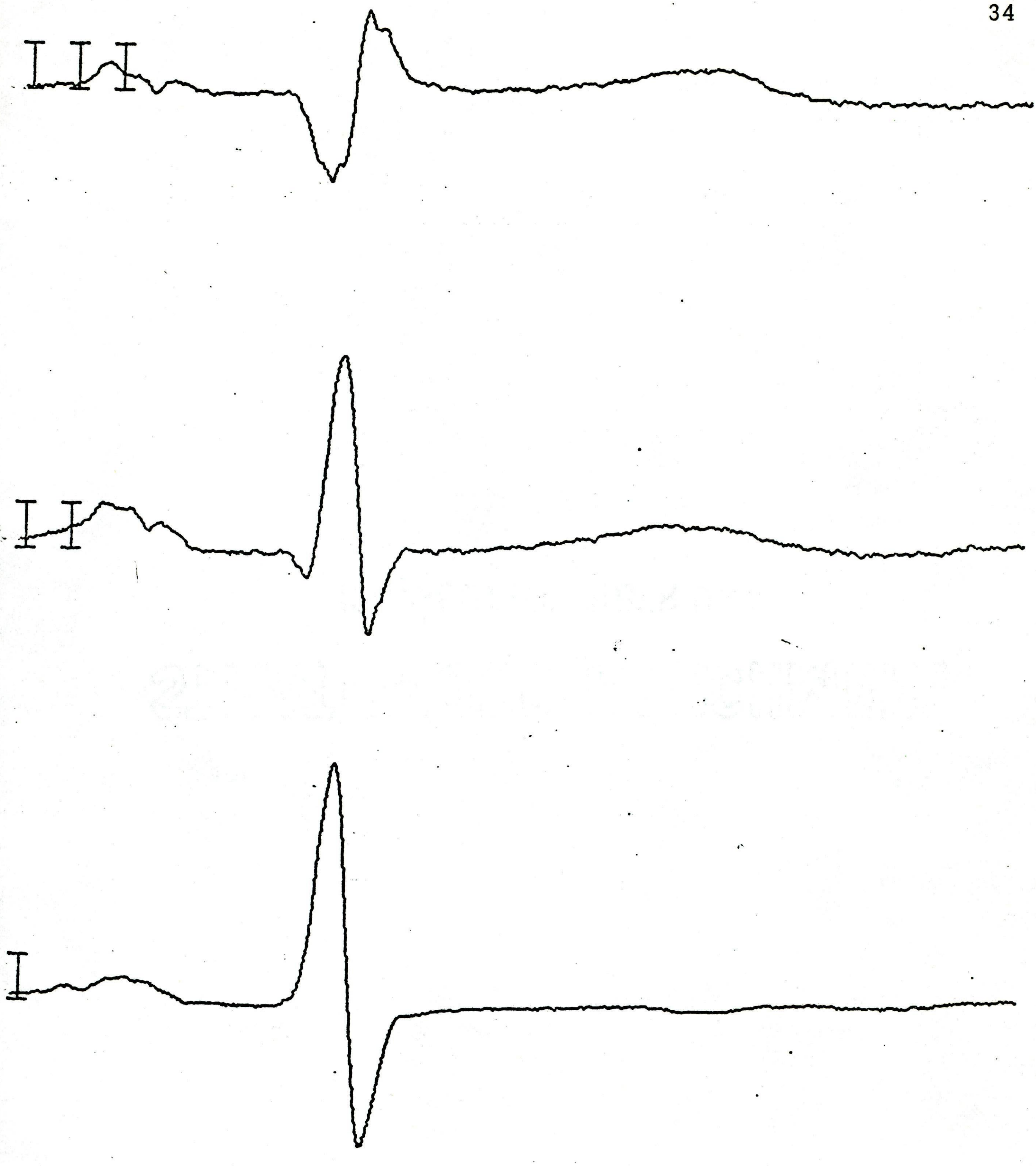


Fig. 20

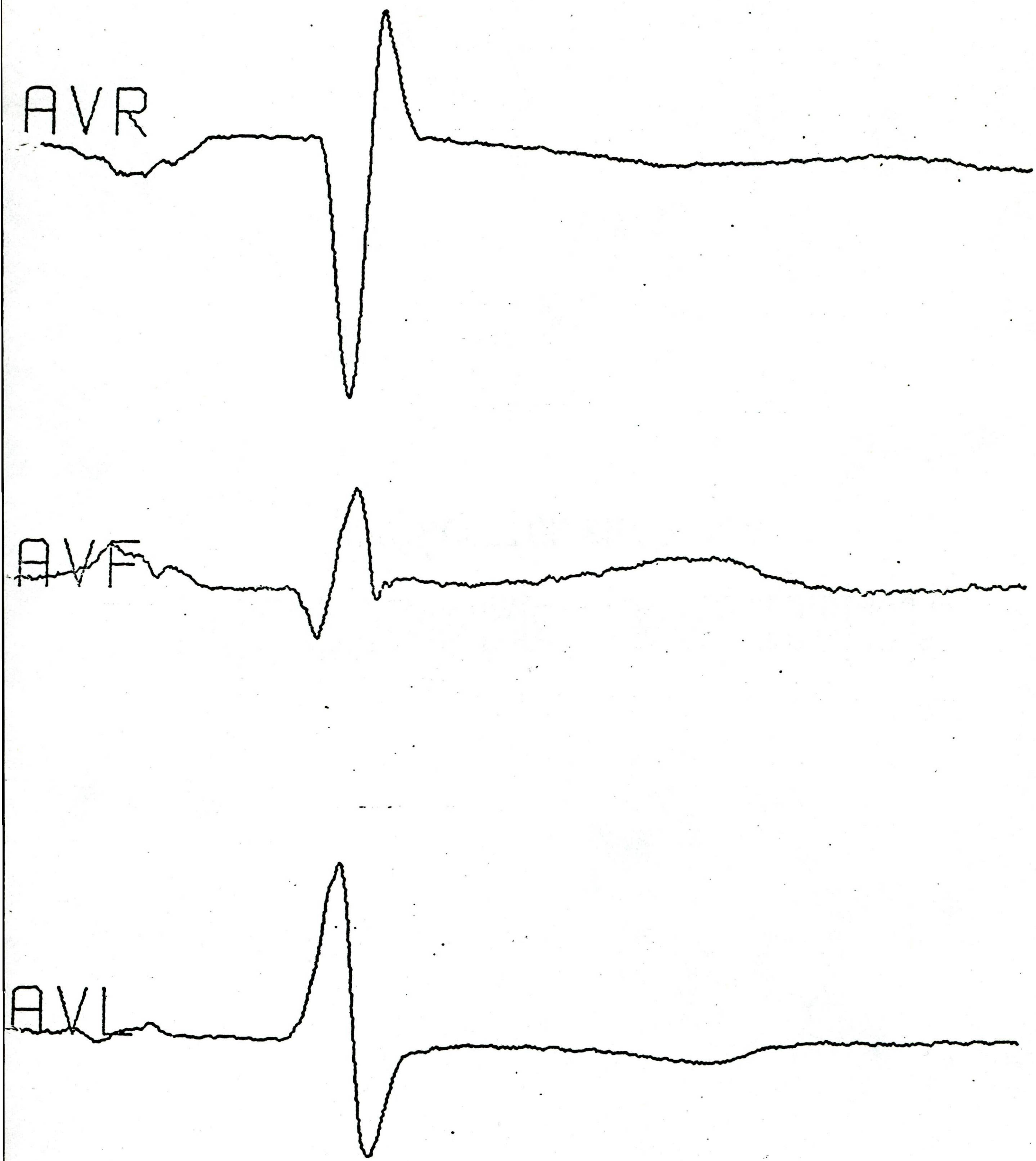


Fig. 20

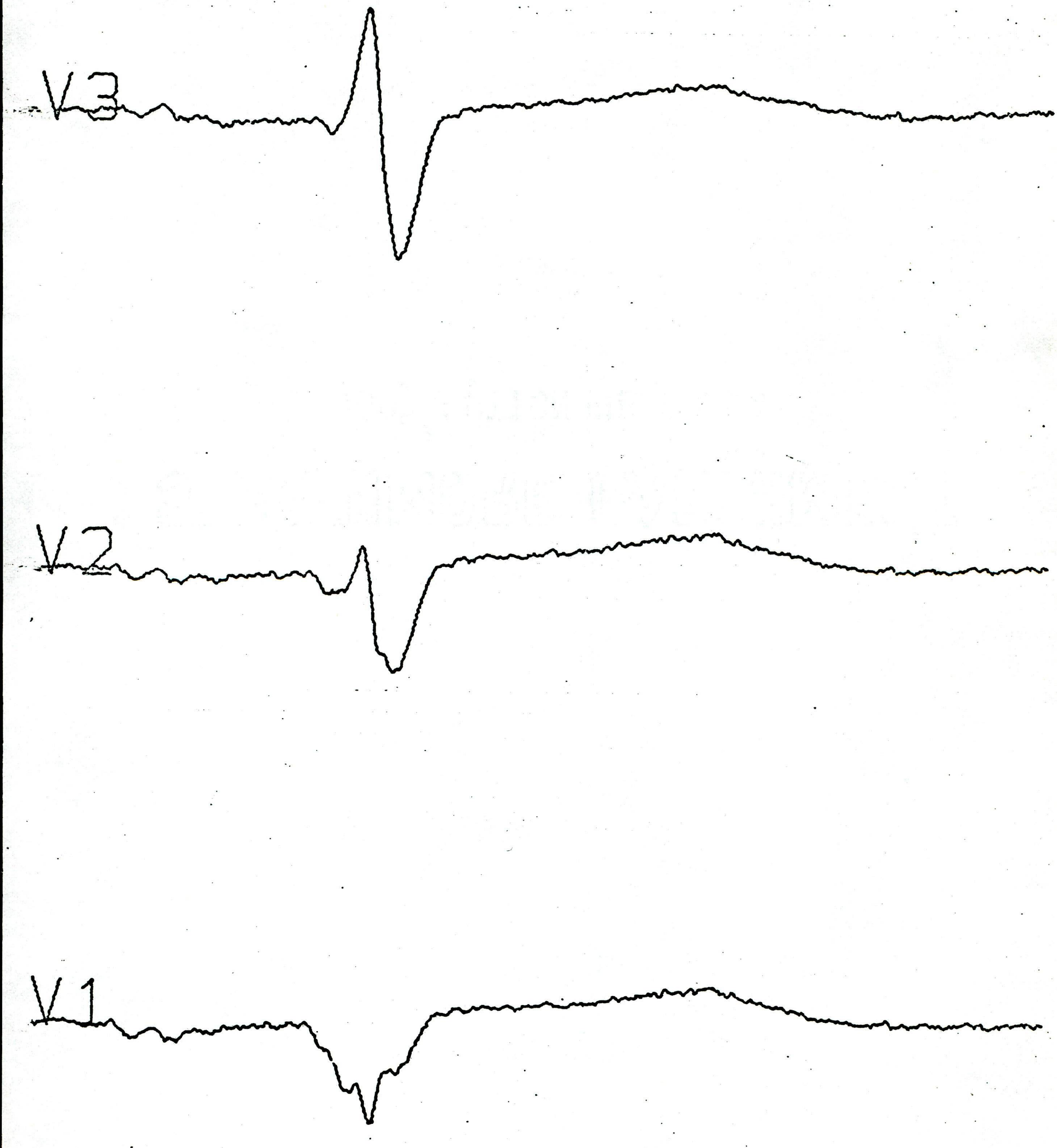


Fig. 20

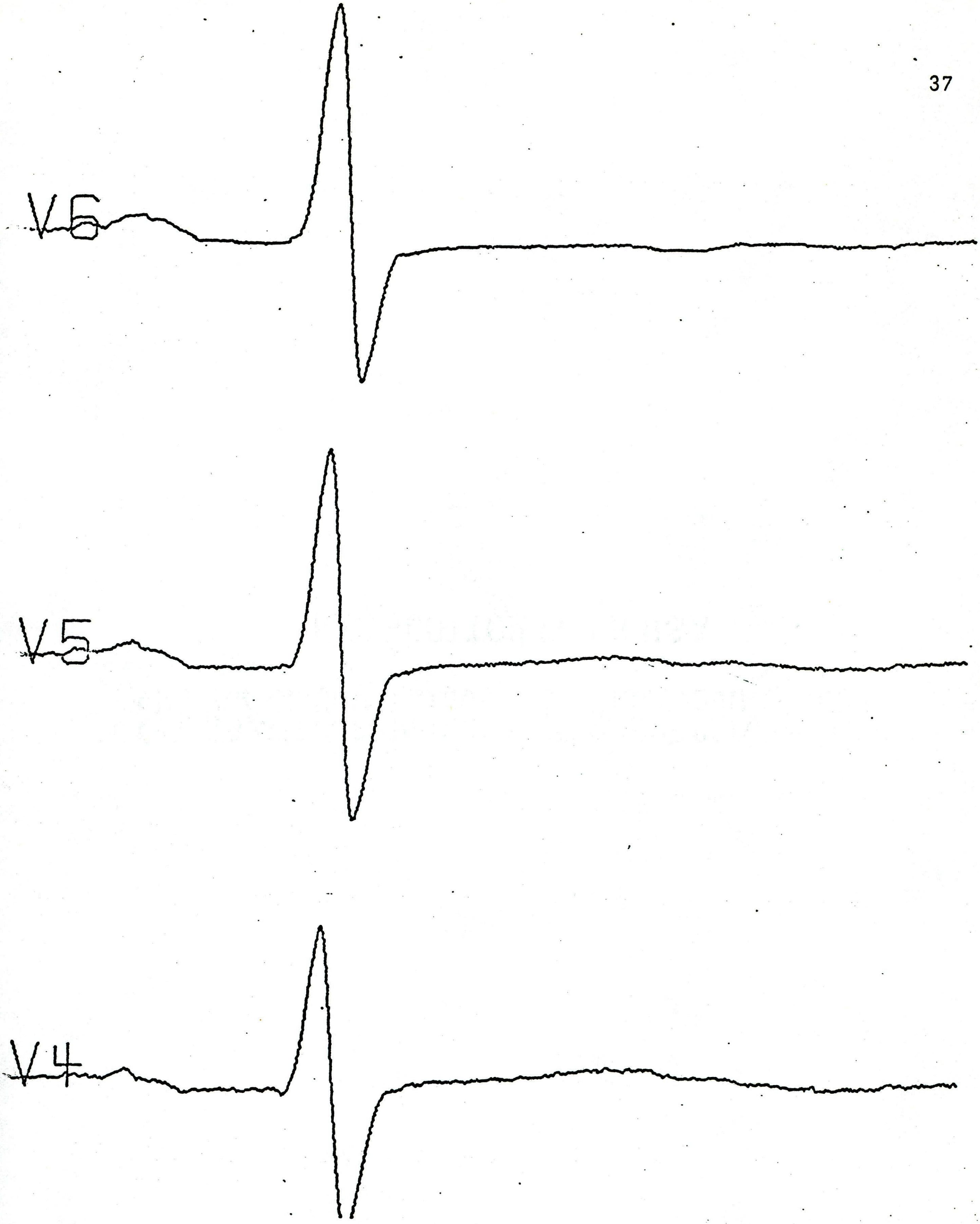


Fig. 20

CHAPTER 6

POLARCARDIOGRAMS

The subject of polarcardiography has been found in literature since 1961, but has not been used extensively as a diagnostic tool due to the lack of information about it. The polarcardiogram does not generate new information but can bring out relationships not previously suspected. Once aware of these new relationships not previously used, physicians will have another tool for diagnosis.[2]

The polarcardiogram is a three-dimensional representation of the direction of the electrical activity in the heart; however, it does not give the magnitude of this electrical activity.

In order to present a three-dimensional representation of a polarcardiogram on a two-dimensional plane, it is necessary to consider a body as a sphere made of a substance that is everywhere equal in conductance of electrical current, with the heart at the center. Then this three-dimensional sphere can be projected onto a two-dimensional plane. This presents the problem of seeing only one side of the sphere at a time. This difficulty can be overcome by the use of the Aitoff projection to display the polarcardiogram, which projects the entire sphere (front as well as back) onto a two-dimensional plane. See Figure 21, an Aitoff projection.

The Aitoff projection is given by latitude and longitude. The latitude goes from -90 (bottom of the body) to 90 (top of the body). The longitude goes from -180 to 180 with +180 projecting the back of the body and 0, the front. Therefore the point on a sphere can be mapped onto a point on the Aitoff projection. Aitoff's formulas take a point on the

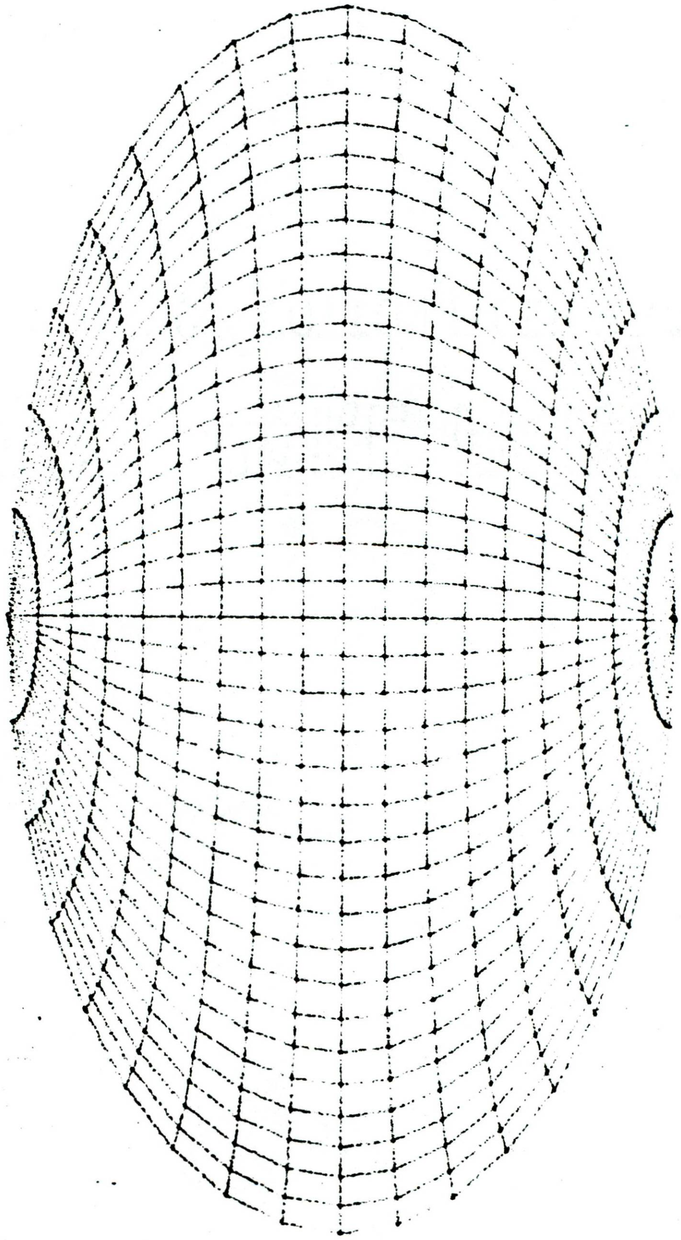


Fig. 21

sphere with latitude and longitude coordinates to produce a new point for the Aitoff projection with X and Y coordinates. The equations are as follows:

$$X = \cos(Q) * \sin(\Theta/2)$$

$$Y = \sin(Q)$$

$$S = X^2 + Y^2$$

$$r = \sqrt{2 * (1 - \sqrt{1 - 5}}$$

$$S = r \cdot \sqrt{S}$$

$$X = X * S$$

$$Y = Y * S / 2$$

where Q = longitude
Θ = latitude

Another form of the above equation is as follows:

$$X = \cos(Q) * \sin\left(\frac{\Theta}{2}\right) * \left(\frac{\sqrt{2 * \left(1 - \sqrt{1 - \left[(\cos(Q) * \sin(\Theta/2)^2 + \sin(Q)^2\right]}\right)}}{\sqrt{\cos(Q) * \sin(\Theta/2)^2 = (\sin(Q))^2}} \right)$$

$$Y = \sin(Q) * \left(\frac{\sqrt{2 * \left(1 - \sqrt{1 - \left[(\cos(\Theta) * \sin(\Theta/2)^2 + (\sin(\Theta))^2\right]}\right)}}{\sqrt{\cos(Q) * \sin(\Theta/2)^2 + (\sin(\Theta))^2}} \right)$$

The programs for the Aitoff projection in Basic on the Tektronix terminal and in Fortran on the EMR 6130 are in Appendix 4, Examples 1 and 2.

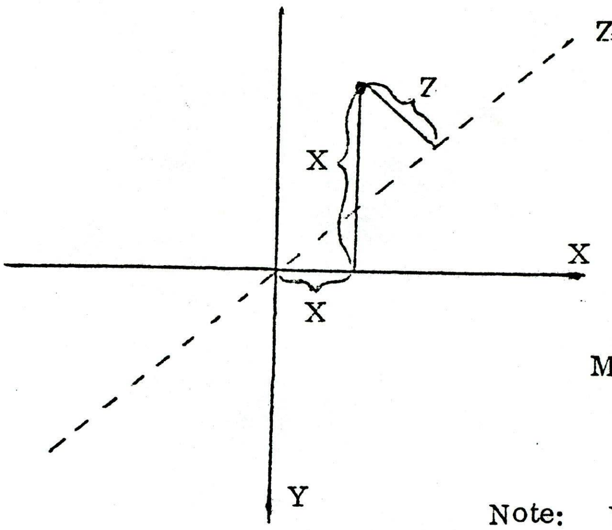
The data for the polarcardiogram comes from the Frank leads. The data from the Frank leads is digitized by an A to D (analog to digital) converter. These data are sampled at a rate of 250 per second. Therefore, the electrical current of the heart is represented by a set of points, each point a 250th of a second from the next point. Each point is located in space by an X, Y and Z coordinate. Since the polarcardiogram is a tracing of the direction of the electrical current in the heart onto the Aitoff projection, it is necessary to do a transformation of coordinate systems from X, Y, Z coordinates to Aitoff's coordinates. Then each point, when given Aitoff's coordinates, may be mapped onto the Aitoff projection. The transformation of coordinates from X, Y, Z coordinates to Aitoff's coordinates is as follows:

1. First, the X, Y, Z coordinates must be transformed into spherical coordinates with θ, δ and M. See Fig. 22.

$$M = \sqrt{X^2 + Y^2 + Z^2}$$

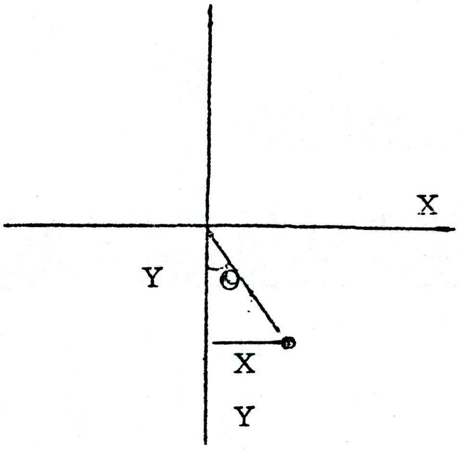
$$\theta = \mathrm{TAN}^{-1} \left(\frac{X}{-Y} \right)$$

$$\delta = \mathrm{TAN} \left(\frac{X}{Z} \right)$$



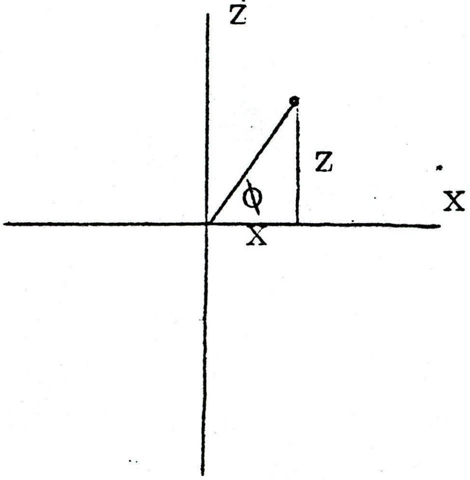
$$M = \sqrt{X^2 + Y^2 + Z^2}$$

Note: Y is negative.



$$\mathrm{TAN}\ (\Theta) = \frac{X}{-Y}$$

$$\Theta = \mathrm{TAN}^{-1}\left(\frac{X}{-Y}\right)$$



$$\mathrm{TAN}\ (\phi) = \frac{X}{Z}$$

$$\phi = \mathrm{TAN}^{-1}\left(\frac{X}{Z}\right)$$

Fig. 22

2. Next, the spherical coordinates M, Θ, ϕ, must be transformed into Aitoff's coordinates. This is done by letting Θ and ϕ be the latitude and longitude in the previous Aitoff equations.

The M is not used since M is magnitude, and the polarcardiogram does not show magnitude. The Θ and ϕ indicate direction and are therefore necessary to the polarcardiogram. M (magnitude) is important in another way since it must be observed that when M is "relatively small" or close to zero, there is very little electrical activity of the heart. Other sources of electrical activity can be detected by the leads. These are disturbances which will show up as small changes in the X, Y, and Z coordinates of each point. These small changes in the X, Y, and Z coordinates will cause drastic changes in direction, not magnitude.

Since these drastic changes in direction are caused by various disturbances rather than by cardiac electrical activity, they are not plotted on the polarcardiograms. These disturbances are called "noises" and if plotted would show up as lines going in all directions without any pattern. Therefore these "noises" are not plotted on the polarcardiogram.

Since "noise" is most critical when the magnitude is relatively small, it can be eliminated by not plotting any points on the polarcardiogram when the magnitude is less than some arbitrary value. In order to provide a polarcardiogram with the balance between the amount of information presented on the polarcardiogram versus the amount of "noise" eliminated, the value of the magnitude used for "noise" elimination should be chosen so that at least parts of the P complex, QRS complex, and T complex will be plotted and nothing else. (See Fig. 23.)

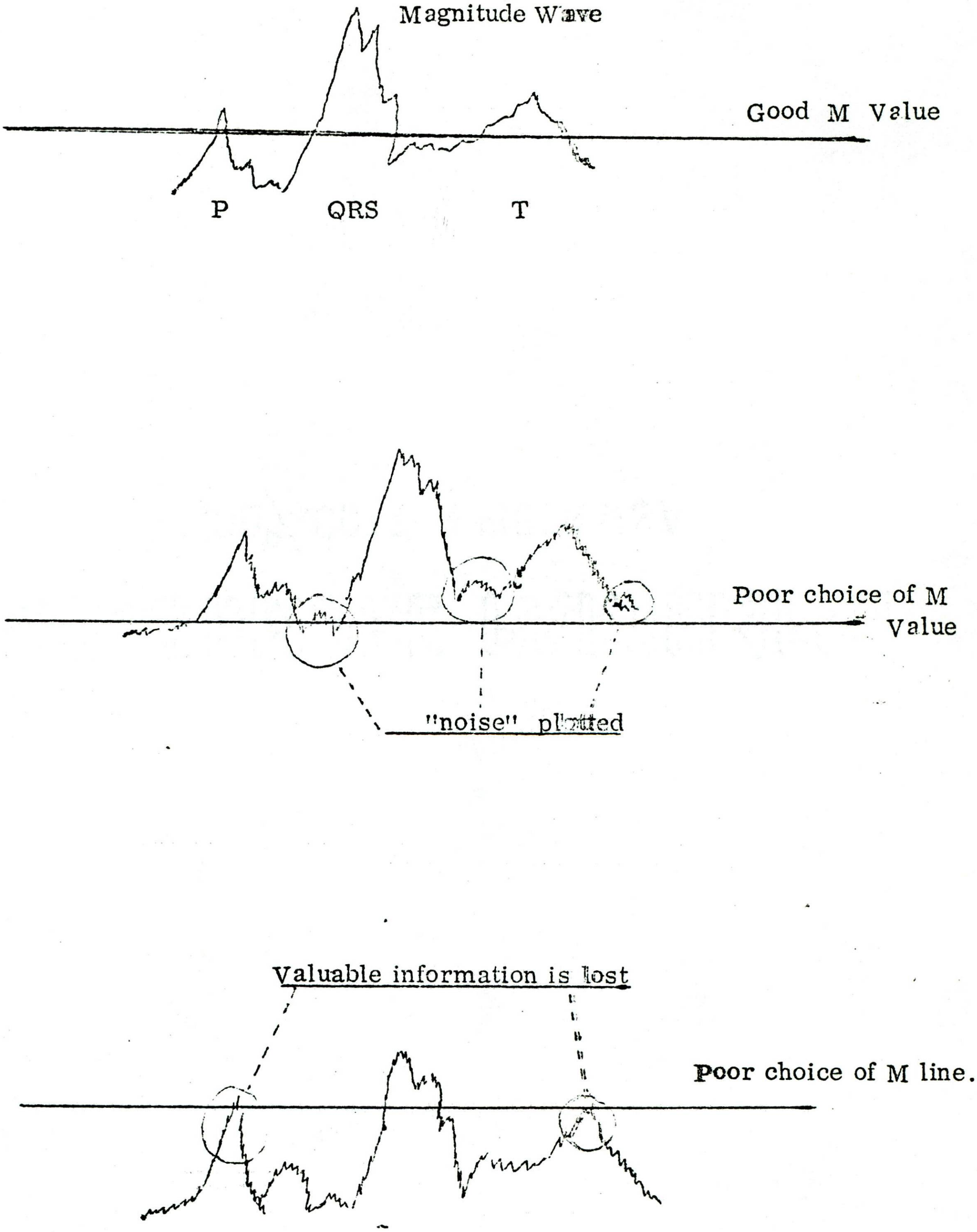

Magnitude Wave
Good M Value
P
QRS
T
Poor choice of M Value
"noise" plotted
Valuable information is lost
Poor choice of M line.

Fig. 23

(See Fig. 24 and 25.) Figure 24 is a polarcardiogram produced by a Basic program on the Tektronix terminal and Figure 25 is a polarcardiogram produced by a Fortran program on the EMR 6130 computer.

The programs for plotting the polarcardiograms are in Appendix 4, Examples 2 and 3.

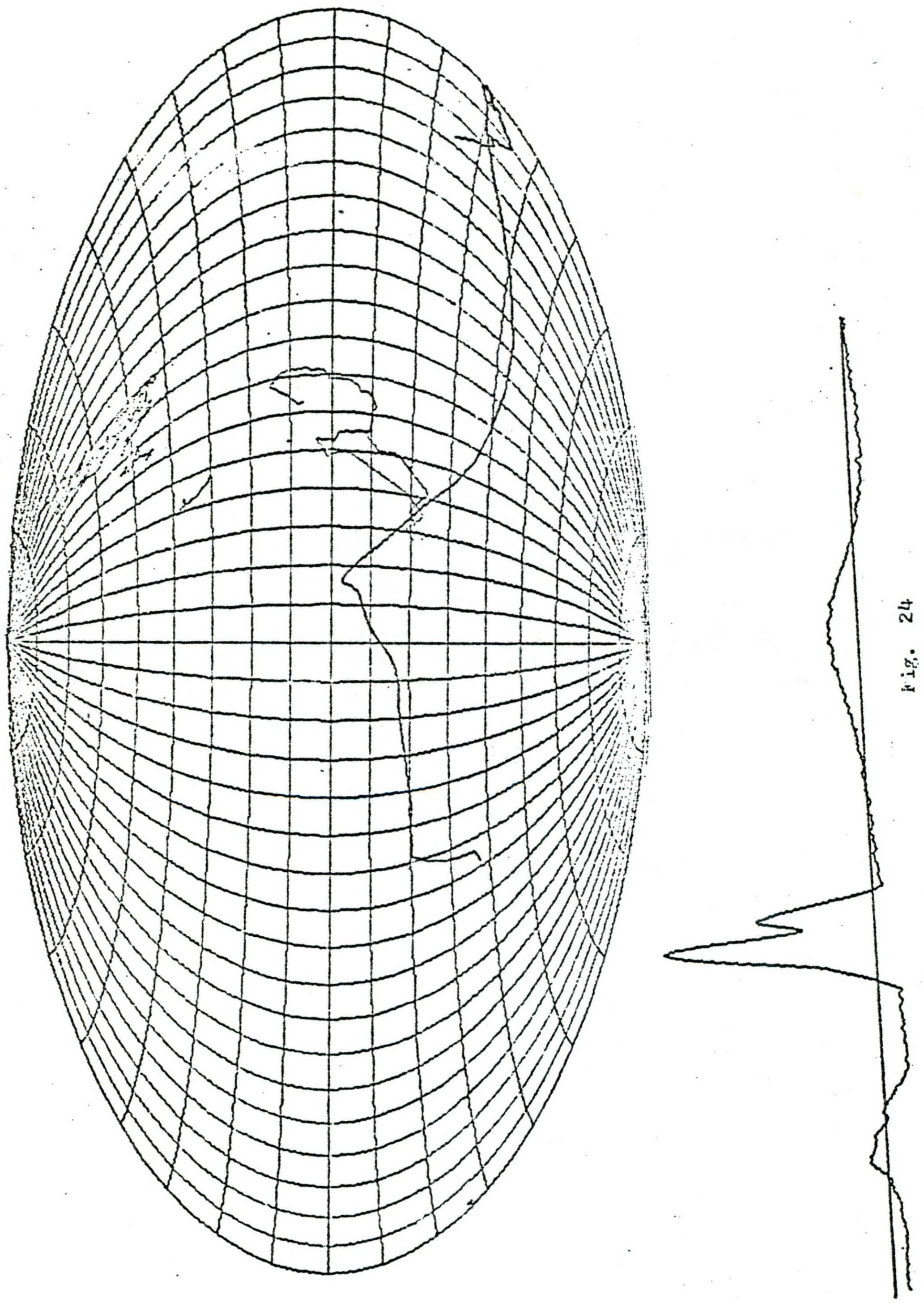


Fig. 24

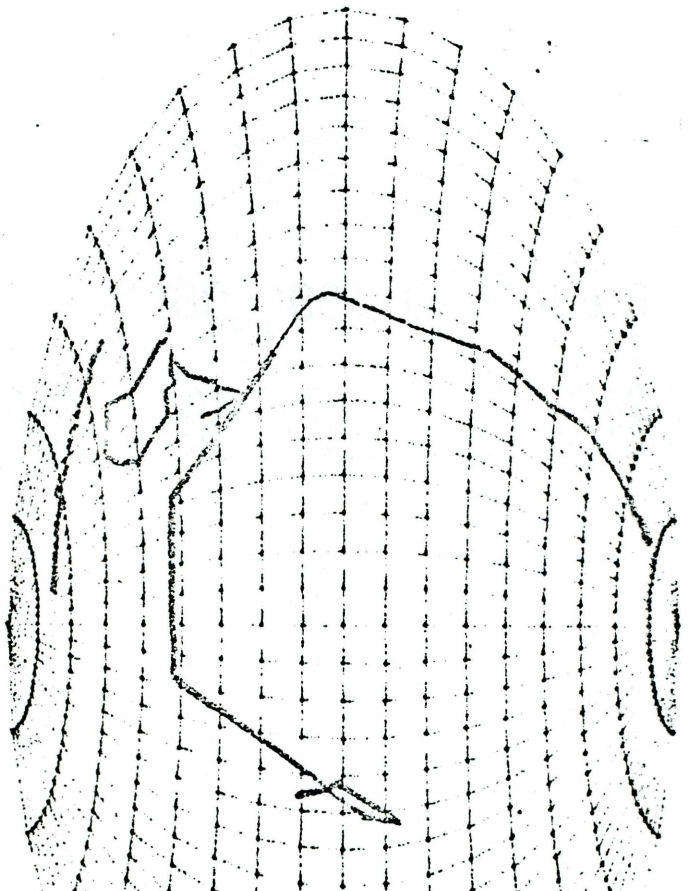


Fig. 25

CHAPTER 7

DIAGNOSTIC CRITERIA BY MAGNITUDE VS. ANGLE WAVES

Dr. Gordon Dower has derived polarcardiographic criteria for diagnosis of myocardial infarction using the transverse magnitude vs. beta wave for the diagnosis of anterior myocardial infarction, and the sagittal magnitude vs. gamma wave for the diagnosis of inferior myocardial infarction.

The programs for producing the transverse magnitude wave, beta wave, sagittal wave and gamma wave were programmed. Then the criteria for myocardial infarction derived by Dr. Dower were used to determine whether 20 patients were normal or whether they had either anterior or inferior myocardial infarction. The results were then compared with the angiograms of these patients. The EKG diagnosis is an interpretation of a single cardiologist. This interpretation may vary from one cardiologist to the next. The angiogram was used as the gold standard. The results of this test are in Figure 26.

The results show that the magnitude vs. angle waves are quite accurate for diagnosis of myocardial infarction. There were no false negatives and only 4 false positives. The false positives indicate myocardial infarction for a patient that has an angiogram that shows no infarction. This indicates that the test may be slightly too sensitive, but very useful as a tool to confirm myocardial infarction or to indicate myocardial infarction. Dr. Dower's criteria for myocardial infarction are given in Figure 27.

Patient	EKG Diagnosis	Diagnosis Based On The Angiogram	Anterior	Inferior
1.	N	N	−	−
2.	N	NA	−	−
3.	N	N	−	−
4.	N	N	(+)	−
5.	I	I	−	+
6.	I	I	−	+
7.	I	NA	−	+
8.	I	IP	−	+
9.	AI	NA	+	+
10.	AI	AI	+	+
11.	A	A	+	(+)
12.	A	NA	+	−
13.	A	A	+	(+)
14.	A	A	+	−
15.	A	A	+	−
16.	PI	I or N	−	?
17.	PI	PI	(+)	+
18.	I	I	−	+
19.	I	I or L	+	+
20.	AI	N?	+	+ ←

N Normal
I Inferior
A Anterior
NA Not Available
L Lateral
P Posterior
O False Positive
← Non-conclusive Diagnosis

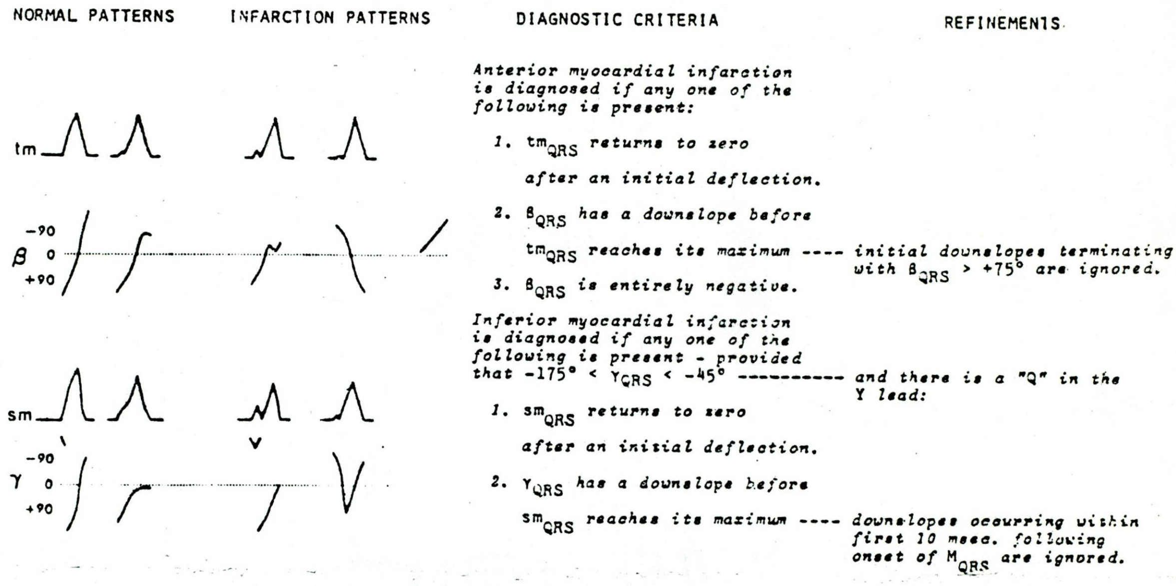


Fig. 27

(Reprinted from Dower, G.E., Polarcardiography, pg 86, 1971)

The criteria used on the patients in this study were the same except for the beta wave. The beta wave in this study was graphed with +90 being "up" or positive on the graph and -90 being "down" or negative on the graph. Therfore, the criteria of slope for the beta wave are the opposite in sign to that of Dr. Dower's criteria for the beta wave. Thus the criteria for myocardial infarction are as follows:

1. If any of the following is present then the diagnosis is Anterior Myocardial Infarction.
   - a.) TM_{QRS} returns to zero after an initial deflection.
   - b.) $Beta_{QRS}$ has an upward slope before TM_{QRS} reaches its maximum. (Initial upward slopes terminating with $Beta_{QRS}$ greater than $+75^o$ are ignored)
   - c.) $Beta_{QRS}$ is entirely positive.
2. If any one of the following is present, provided that -175^o is less than gamma which is less than -45^o and there is a "Q" in the Y lead.
   - a.) SM_{QRS} returns to zero after an initial deflection.
   - b.) $Gamma_{QRS}$ has a downward slope before SM_{QRS} reaches its maximum. (Down slopes occuring within the first 10 msec. following onset of M_{QRS} are ignored.)

The transverse magnitude wave (TM) is $\sqrt{X^2 + Z^2}$. The sagittal magnitude, (SM) is $\sqrt{Y^2 + Z^2}$. Beta and gamma are polar coordinates of the heart vector in the transverse and sagittal planes. The polarcardiogram contains the data for the beta and gamma waves which are graphed against time.

The procedures for analyzing the results of the magnitude vs. angle wave test for myocardial infarction are as follows:

1. Plot the magnitude vs. angle waves. (See Ap. 4, Ex. 8.)
2. Locate the starting and ending points of the QRS complex in the magnitude waves.
3. Draw a vertical line through the starting and ending points of the QRS complex in the magnitude waves.
4. Analyze the magnitude waves between the vertical lines to detect a return of the magnitude to the base line.
5. Analyze the angle waves between the vertical lines for the slope criteria as described in Figure 27.
6. Make the diagnosis.

The example of the magnitude vs. angle waves in Appendix 4, Example 8, is of a patient that has an inferior infarction.

Infarctions disturb the QRS vectors in two ways which lead to the criteria for myocardial infarction in the magnitude vs. angle waves.

The two ways that the infarction disturbs the QRS vectors are:

1) disturbs the normal pathway of ventricular activation and

2) imbalances the vectoral forces.

Since the infarction causes less electrical activity in the area where the infarction is located, the other areas of the heart have what appears to be exaggerated electrical activity. This would cause the electrical activity to be at a different direction angle and magnitude than that of a normal heart. The polarcardiogram and specifically the magnitude vs. angle waves show this difference in direction, angle and magnitude much better than the scalar EKG which do not directly show angle. Therefore, the polarcardiogram and the magnitude vs. angle waves are another useful tool for the cardiologist for the diagnosis of myocardial infarction.

BIBLIOGRAPHY

1. Lawrence E. Lamb, M.D., Your Heart and How to Live With It. New American Library 1975 pp. 51,58

2. Gordon Ewbank Dower, M.D., B.S., F.A.C.B., Polarcardiography, Charles C. Thomas, Publisher. 1971 pp. 13,14.

3. Gordon Ewbank Dower, and Robert A. Bruce, Journal of Progress in Cardiovascular Dinasi, Vol. XIX No. 1 July 76. Article: Progress in Polarcardiography.

APPENDIX	I	Programs for Transfering Data Between Computers.
APPENDIX	II	Vector Loops --Programs and Illustrations.
APPENDIX	III	Twelve-Lead Synthesis Programs and Illustrations.
APPENDIX	IV	Polarcardiography Programs and Illustrations.

APPENDIX I

Ex. 1	MIIS Program: Transfers Data from the 6130 to MIIS.
Ex. 2	MIIS Program: Transfers Data from MIIS to the Tektronix terminal.
Ex. 3	Tektronix Basic Program: Transfers data from MIIS to the Tektronix terminal.
Ex. 4 a.	EMR 6130 Fortran Program: Reads patient data from Dower Cart.
Ex. 4 b.	EMR 6130 Fortran Program: Produces a Median Wave.
Ex. 5	EMR 6130 Fortran Program: Transfers Data from the 6130 to MIIS.
Ex. 6	Median Waves--Produced by a Fortran Program on the EMR 6130.
Ex. 7	Median Waves--Produced by a Basic Program on the Tektronix terminal.
Ex. 8	Diagram of the Outline of the Thesis.

APPENDIX II

Ex. 1	EMR 6130 Fortran Program--Produces Vector Loops.
Ex. 2	Vector Loops--Produced by a Fortran Program on the EMR 6130.
Ex. 3	Tektronix Basic Program--Produces Vector Loops.
Ex. 4	Vector Loops--Produced by a Basic Program on the Tektronix terminal.

APPENDIX III

Ex. 1	EMR 6130 Fortran Program--Synthesizes Twelve Leads.
Ex. 2	Twelve Lead Synthesis--Produced by a Fortran Program on the EMR 6130.
Ex. 3	Tektronix Basic Program--Synthesizes Twelve Leads.
Ex. 4	Twelve Lead Synthesis--Produced by a Basic Program on the Tektronix terminal.
Ex. 5	Tektronix Basic Program--Synthesizes Twelve Leads. (Dower Methods)
Ex. 6	Twelve Lead Synthesis (Dower Method)--Produced by the Tektronix terminal.

APPENDIX IV

Ex. 1	Tektronix Basic Program--Produces Aitoff Projection.
Ex. 2	Aitoff Projection--Produced by a Basic Program on the Tektronix terminal.
Ex. 3	EMR 6130 Fortran Program--Produces Aitoff Projection.
Ex. 4	Polar Plot on Aitoff Projection--Produced by a Fortran Program on the EMR 6130.
Ex. 5	Tektronix Basic Program--Produces Polar Plot.
Ex. 6	Polar Plot--Produced by a Basic Program on the Tektronix terminal.
Ex. 7	Polar Plot on Aitoff Projection--Produced by a Basic Program on the Tektronix terminal.
Ex. 8	Magnitude vs. Angle Plot.
Ex. 9	Magnitude vs. Angle Program--Produced by a Basic Program on the Tektronix terminal.

APPENDIX I

```
WXYZ ;SEND X,Y,Z TO 4051
     S M=1
L1   S XYZ=^XYZ(M),M=M+1 I $E(XYZ,1,1)=0 S Z=M-2 G L1+2
     G L1
     F T=1:1:Z S XYZ=^XYZ(T) F I=1,5,9 R X W $E(XYZ,I,I+3),!
     F N=1:1:3 R X W "0",!
     Q
```

EX. 1 MIIS PROGRAM
TRANSFERS DATA FROM EMR 6130 TO MIIS

```
RXYZ ;READS XYZ TRIPS FROM 6 1 3 0
     S I=1
L1   A 12:24 R XYZ A 0 W XYZ,!
     S ^XYZ(I)=XYZ,I=I+1 G:$E(XYZ,1,4)'=0 L1 U 12 Q
```

EX. 2 MIIS PROGRAM
TRANSFERS DATA FROM MIIS TO TEKTRONIX

```
10 FIND 2
20 DIM X(300),Y(300),Z(300)
30 I=1
40 PRINT @40:"XXXXXXXX"
50 INPUT @40:X(I)
60 PRINT X(I)
70 PRINT @40:"YYYYYYYY"
80 INPUT @40:Y(I)
90 PRINT @40:"ZZZZZZZZ"
100 INPUT@40:Z(I)
110 IF X(I)=0 THEN 140
120 I=I+1
130 GOTO 40
140 FOR J=1 TO I
150 PRINT @33:X(J),Y(J),Z(J)
160 NEXT J
170 END
```

EX. 3 TEKTRONIX BASIC PROGRAM
TRANSFERS DATA FROM MIIS TO TEKTRONIX

```
C       FRED/READ PATIENT DATA AND STORE IN PEFILE:PCGDAT
C
        INTEGER SUCK(100)
        INTEGER RTD(3,3072,2),NAME(10),LIMP(128),BLS(3),DAD(2)
        DIMENSION SFS(3)
        EQUIVALENCE (PO,$0),(NAME,LIMP),(BLS,LIMP(11)),(SFS,LIMP(22))
        EQUIVALENCE (SENS,LIMP(14))
        DATA DAD/0,0/
        CALL GETFIL ('PCGSFS')
        CALL CIO (27,6,PO,0,IST,128,LIMP,DAD)
5       IF (IST .EQ. 0) GO TO 5
        WRITE (17,5246) BLS,SFS
5246    FORMAT (3(I5/),3(F12.6/))
1       WRITE (17,10)
10      FORMAT (' ENTER NAME')
        READ (15,20) NAME
20      FORMAT (10A2)
        WRITE (17,41)
41      FORMAT (' ENTER SENSITIVITY')
        READ (15,602) SENS
602     FORMAT (F12.5)
        ISN=2./SENS
        WRITE (17,30)
30      FORMAT (' SEND DATA')
        CALL PEFILE (1,289,126,IPT,'PCGDAT')
        CALL GETFIL ('PCGDAT')
        CALL CIO (27,6,PO,2,IST,128,LIMP,DAD)
40      IF (IST .EQ. 0) GO TO 40
        DAD(2)=1
        DO 100 LL=1,2
/I
        EQP   38
/F
        CALL CIO (38,5,PO,104,IST,18432,RTD)
        IF (IST .EQ. -32764) GO TO 1
        DO 90 I=1,2
        DO 70 J=1,3072
        IF (RTD(1,J,I) .EQ. -9999) GO TO 1
        DO 60 K=1,3
60      RTD(K,J,I)=RTD(K,J,I)*ISN/2-BLS(K)
70      CONTINUE
80      IF (IST .EQ. 0) GO TO 80
        CALL CIO (27,6,PO,2,IST,9216,RTD(1,1,I),DAD)
        DAD(2)=DAD(2)+72
90      CONTINUE
100     CONTINUE
        CALL BEEP
107     CALL CIO (38,5,PO,104,IST,100,SUCK)
```

EX. 4A 6130 FORTRAN PROGRAM
READS DATA FROM THE DOWER CART

```
      IF (IST .NE. -32764) GO TO 107
      WRITE (17,110)
110   FORMAT (' TURN OFF DATA SWITCH')
120   IF (IDSCQM .EQ. 0) GO TO 120
      CALL LINK ('MEDIAN')
      END
      EOF
```

```
      TTL   MEDIAN
      IMPLICIT INTEGER (A-Z)
      LOGICAL REDO
      INTEGER MAG(9984),BUF(3,768),LIMP(128),PEAK(25),CLAMPP(25)
      INTEGER CLAMPV(3,25)
      REAL FOX(3),MF
      REAL SLAP(3)
      REAL ACCEPT
      REAL SUM,FIX(3)
      REAL SLIP
      REAL SENS
      DIMENSION MAXMAG(13)
      DIMENSION LIPE(13)
      DIMENSION AH(13),GG(13)
      DIMENSION NAME(10)
      DIMENSION MXYZ(20,701)
      DIMENSION DAD(2)
      DIMENSION ACLAM(3)
      DIMENSION RR(25),CLAM(25),TCLAMP(3)
      COMMON MXYZ
      EQUIVALENCE (TCLAMP(4),CLAMPV)
      EQUIVALENCE (MXYZ,MAG)
      EQUIVALENCE (NAME,LIMP),(SENS,LIMP(14))
      EQUIVALENCE (FIX,LIMP(22))
      EQUIVALENCE (PO,$0)
      DATA DAD/0,0/
      DATA REDO/.FALSE./
      DATA BND,EDD/250,450/
      DATA CLOMP/75/
      WRITE (17,1701)
1701  FORMAT (' ENTER FILE NUMBER')
      NPASS=ACCEPT(15)
C
C     FIND MAXIMUM MAGNITUDE
C
922   CONTINUE
1210  CONTINUE
1215  CALL RPCG (0,LIMP,1)
      MF=AMIN1(FIX(1),FIX(2),FIX(3))
      DO 1073 I=1,3
1073  FOX(I)=FIX(I)/MF
      G=1
      MINMAG=20000
      DO 1260 I=1,13
      MAXMAG(I)=0
      LIPE(I)=0
      CALL RPCG ((I-1)*18+1,BUF,18)
      DO 1250 J=1,768
      IF (G .GT. 9984) GO TO 8102
      SUM=0.0
      IF (.NOT. REDO) GO TO 1523
      IF (G .NE. CLAMPP(MPEAK)) GO TO 989
      CLUMP=CLAMPP(MPEAK)
```

```
      IF (MPEAK .NE. NPEAK) GO TO 2021
      DO 1847 L=1,3
      ACLAM(L)=CLAMPV(L,MPEAK)
1847  SLAP(L)=(TCLAMP(L)-CLAMPV(L,NPEAK))/FLOAT(MEDRR)
      GO TO 989
2021  EMS=CLAMPP(MPEAK+1)-CLAMPP(MPEAK)
      DO 8011 L=1,3
      ACLAM(L)=CLAMPV(L,MPEAK)
8011  SLAP(L)=(CLAMPV(L,MPEAK+1)-CLAMPV(L,MPEAK))/FLOAT(EMS)
      MPEAK=MPEAK+1
989   DO 1989 L=1,3
1989  SUM=SUM+((BUF(L,J)-ACLAM(L)-(G-CLUMP)*SLAP(L))*FOX(L))**2
      GO TO 1620
1523  DO 1240 L=1,3
1240  SUM=SUM+(BUF(L,J)*FOX(L))**2
1620  MAG(G)=SQRT(SUM)
      IF (MAG(G) .LE. MAXMAG(I)) GO TO 762
      MAXMAG(I)=MAG(G)
      GG(I)=G
762   CONTINUE
      IF (MAG(G) .LT. MINMAG) MINMAG=MAG(G)
      IF (J .LT. 21) GO TO 1250
      LID=MAG(G)-MAG(G-20)
      IF (LID .GT. LIPE(I)) LIPE(I)=LID
1250  G=G+1
1260  CONTINUE
8102  CONTINUE
      DO 124 I=1,13
      SUM=0.0
      GM=GG(I)-20
      GP=GG(I)+20
      DO 123 J=GM,GP
123   SUM=SUM+MAG(J)
124   AH(I)=SUM/41.
      DO 1810 I=2,7
      DO 1810 J=I,13
      IF (AH(J) .GE. AH(I-1)) GO TO 1801
      K=AH(J)
      AH(J)=AH(I-1)
      AH(I-1)=K
1801  CONTINUE
      IF (LIPE(J) .GE. LIPE(I-1)) GO TO 1805
      K=LIPE(J)
      LIPE(J)=LIPE(I-1)
      LIPE(I-1)=K
1805  CONTINUE
      IF (MAXMAG(J) .GE. MAXMAG(I-1)) GO TO 1810
      K=MAXMAG(J)
      MAXMAG(J)=MAXMAG(I-1)
      MAXMAG(I-1)=K
1810  CONTINUE
      TOPLIM=.60*(MAXMAG(7)-MINMAG)+MINMAG
      ATOP=AH(7)*.15
```

```
      ITOP=MAXMAG(7)*.15
      LOPE=LIPE(7)/2
C
C     FIND CLAMPING POINTS AND THEIR VALUES
C
      NPEAK=0
      WHAT=-1
      CALL RPCG (0,LIMP,1)
      DO 100 G=21,9984
      IF (WHAT) 1307,40,1300
1307  IF (MAG(G) .LT. TOPLIM) GO TO 100
      IF (MAG(G)-MAG(G-20) .LT. LOPE) GO TO 100
      M=1
      MAX=0
      WHAT=10
      WHAT=15
      WHAT=25
      GO TO 100
40    IF (MAXP .LE. CLOMP) GO TO 1290
      IF (IABS(MAXMAG(7)-MAX) .GT. ITOP) GO TO 1295
      GM=MAXP-20
      GP=MAXP+20
      SUM=0.0
      DO 427 I=GM,GP
427   SUM=SUM+MAG(I)
      GUM=SUM/41.
      IF (IABS(AH(7)-GUM) .GT. ATOP) GO TO 1295
      NPEAK=NPEAK+1
      PEAK(NPEAK)=MAXP
      CLAMPP(NPEAK)=MAXP-CLOMP
      I=(CLAMPP(NPEAK)-1)/128*3+1
      CALL RPCG (I,BUF,6)
      H=CLAMPP(NPEAK)-I/3*128
      DO 1280 L=1,3
1280  CLAMPV(L,NPEAK)=BUF(L,H)
1290  M=2
      WHAT=101
      GO TO 100
1295  CONTINUE
      WHAT=-1
      GO TO 100
1300  IF (M .EQ. 2) GO TO 1301
      IF (MAG(G) .LE. MAX) GO TO 1301
      MAX=MAG(G)
      MAXP=G
1301  WHAT=WHAT-M
100   CONTINUE
      DO 406 I=2,NPEAK
406   RR(I-1)=CLAMPP(I)-CLAMPP(I-1)
      MEDRR=MED(RR,NPEAK-1)
      DO 1121 L=1,3
      DO 613 M=1,NPEAK
613   CLAM(M)=CLAMPV(L,M)
```

```
      TCLAMP(L)=MED(CLAM,NPEAK)
      ACLAM(L)=TCLAMP(L)
1121  SLAP(L)=(CLAMPV(L,1)-TCLAMP(L))/FLOAT(MEDRR)
      NPASS=NPASS-1
      IF (NPASS .LE. 0) GO TO 1232
      CLUMP=CLAMPP(1)-MEDRR
      REDO=.TRUE.
      MPEAK=1
      GO TO 922
1232  CONTINUE
      CALL PEFILE (3,80,126,IPT,'SAVEIT')
      CALL FILE (3,1)
      WRITE (4) IFN,NPEAK,PEAK,CLAMPP,MAG
C
C     GATHER QRS COMPLEXES INTO MXYZ FOR SORT
C
      CALL PEFILE (1,1,126*30,IPT,'MEDIAN')
      CALL FILE (1,1)
      IF (MEDRR .GT. 701) GO TO 830
      BND=BND/701.*MEDRR
      EDD=EDD/701.*MEDRR-50.
830   CONTINUE
      ILIM=EDD+BND+1
      BN=1
      ED=NPEAK
      IF (PEAK(1)-BND .LT. 1) BN=2
      IF (PEAK(NPEAK)+EDD .GT. 9984) ED=NPEAK-1
      IPEAK=ED-BN+1
      MBND=-BND
      L=1
      DO 28 I=MBND,EDD
      DO 27 J=BN,ED
      IPJ=I+PEAK(J)
27    CLAM(J)=MAG(IPJ)
      MAG(L)=MED(CLAM(BN),IPEAK)
28    L=L+1
      WRITE (4) ILIM,BND,EDD,CLOMP,MEDRR,SENS,NAME,(MAG(I),I=1,ILIM)
      DO 5000 I=1,3
      G=1
      H=1
1360  CALL RPCG (1,BUF,18)
      DO 1420 J=BN,ED
      JAW=PEAK(J)-BND
      SRW=CLAMPP(J-1)
      IF (J .EQ. 1) SRW=CLAMPP(1)-MEDRR
      ERW=CLAMPP(J)
      EMS=ERW-SRW
      L=EMS-(BND-CLOMP)
      SLUP=CLAMPV(I,J-1)
      IF (J .EQ. 1) SLUP=TCLAMP(I)
      SLIP=(CLAMPV(I,J)-SLUP)/FLOAT(EMS)
1370  DO 1375 H=H,768
      IF (G .EQ. JAW) GO TO 1380
```

```
      G=G+1
      IF (G .GT. 9984) GO TO 42
1375  CONTINUE
      CALL RPCG (-1,BUF,18)
      H=1
      GO TO 1370
1380  CONTINUE
      DO 1410 K=1,ILIM
      IF (K .NE. BND-CLOMP+1) GO TO 1400
      L=0
      SRW=ERW
      ERW=CLAMPP(J+1)
      IF (J .EQ. NPEAK) ERW=SRW+MEDRR
      EMS=ERW-SRW
      SLUP=CLAMPV(I,J)
      SLEP=CLAMPV(I,J+1)
      IF (J .EQ. NPEAK) SLEP=TCLAMP(I)
      SLIP=(SLEP-SLUP)/FLOAT(EMS)
1400  MXYZ(J,K)=BUF(I,H)-SLUP-L*SLIP
      L=L+1
      G=G+1
      IF (G .GT. 9984) GO TO 42
      H=H+1
      IF (H .LE. 768) GO TO 1410
      CALL RPCG (-1,BUF,18)
      H=1
1410  CONTINUE
1420  CONTINUE
42    CONTINUE
C
C     SORT AND EXTRACT MEDIAN WAVE
C
      DO 1460 J=1,ILIM
1460  MXYZ(1,J)=MED(MXYZ(BN,J),IPEAK)
      WRITE (4) ILIM,FIX,(MXYZ(1,J),J=1,ILIM)
5000  CONTINUE
      CALL LINK ('THYME ')
      END
      FUNCTION MED (M,N)
      DIMENSION M(1)
      DO 10 I=2,N
      DO 10 J=I,N
      IF (M(J) .GE. M(I-1)) GO TO 10
      MN=M(J)
      M(J)=M(I-1)
      M(I-1)=MN
10    CONTINUE
      I=N/2
      J=N-2*I
      I=I+J
      J=I+1-J
      MED=(M(I)+M(J))/2
      RETURN
```

END
EOF

```
      TTL    POLK READ CALIBRATION DATA AND STORE IN PEFILE:
      INTEGER DATJT
      INTEGER SPT(3,2)
      INTEGER LIMP(128),DAD(2)
      INTEGER DAT(3,6144),SS(3),YH(3,50),YN(3),DATL(3),YM(4),YL(3,3)
      INTEGER BAT(3,3072,2),RAT(128,145)
      DIMENSION YS(3),SF(3)
      COMMON RAT
      EQUIVALENCE (BAT,DAT,RAT)
      EQUIVALENCE (SF,LIMP(22)),(P0,$0)
      DATA DAD/0,0/
      PAUSE
1     CONTINUE
      DO 10 I=1,3
      YS(I)=0.0
      DO 5 J=1,3
5     YL(I,J)=0
10    SS(I)=1
90    CONTINUE
/I
      EQP   38
/F
      CALL CIO (38,5,P0,104,JST,18432,BAT)
65    CALL CIO (22,6,P0,2,IST,18432,BAT(1,1,1),DAD)
81    IF (IST .EQ. 0) GO TO 81
      IF (JST .EQ. -32764) GO TO 951
      DAD(2)=DAD(2)+144
      GO TO 90
951   CONTINUE
      CALL KILPIG
      DAD(2)=0
      IK=0
95    CALL CIO (22,6,P0,0,IST,18560,RAT,DAD)
803   IF (IST .EQ. 0) GO TO 803
      DAD(2)=DAD(2)+144
      DO 500 I=1,6144
      IF (DAT(1,I) .EQ. -9999) GO TO 3000
      DO 1000 J=1,3
      DATJI=DAT(J,I)
      JSS=SS(J)
      GO TO (110,120),JSS
110   DO 112 K=2,500
      DO 111 L=K,500
      IF (DAT(J,K-1) .LE. DAT(J,L)) GO TO 111
      NEX=DAT(J,K-1)
```

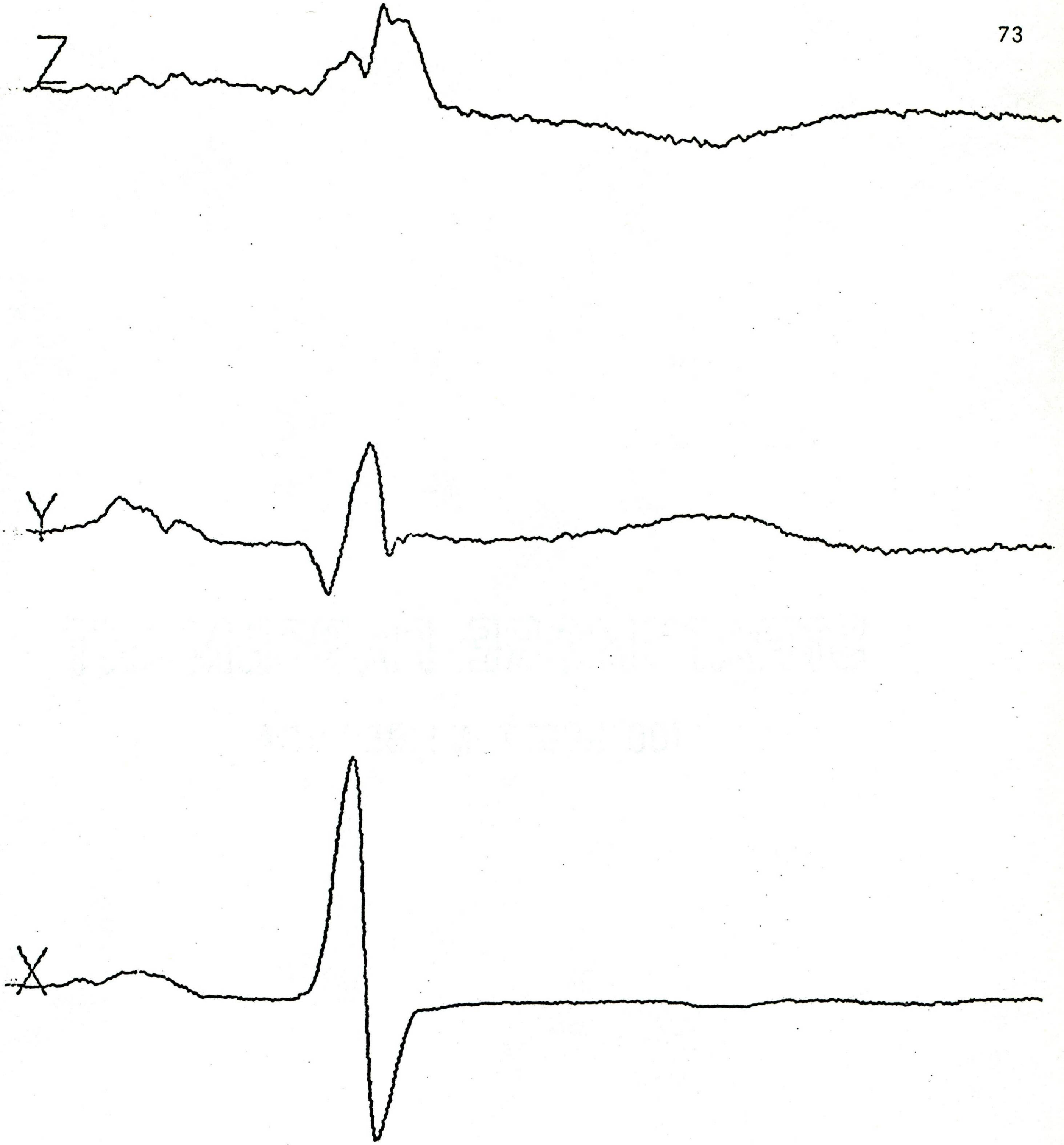
EX. 4B 6130 FORTRAN PROGRAM
PRODUCES A MEDIAN WAVE

```
      DAT(J,K-1)=DAT(J,L)
      DAT(J,L)=NEX
111   CONTINUE
112   CONTINUE
      Z=0.0
      DO 113 K=201,300
113   Z=Z+DAT(J,K)
      YL(J,2)=Z/100.
      SS(J)=2
      I=500
      GO TO 1000
120   IF (IABS(DATJI) .LT. 100) GO TO 1000
      IF (DATJI .GT. 0 .AND. YL(J,3) .EQ. 0) GO TO 140
      IF (DATJI .LT. 0 .AND. YL(J,1) .EQ. 0) GO TO 150
      GO TO 1000
140   CONTINUE
      Z=0.0
      IPN=I+9
      DO 145 K=I,IPN
      IF (IABS(DAT(J,K+1)-DAT(J,K)) .GT. 10) GO TO 1000
      IF (IABS(DAT(J,K+1)-DATJI) .GT. 50) GO TO 1000
      Z=Z+DAT(J,K+1)
145   CONTINUE
      YL(J,3)=Z/10.
      SPT(J,2)=IK+I
      SF(J)=1./(YL(J,3)-YL(J,2))
      GO TO 1000
150   CONTINUE
      Z=0.0
      IPN=I+9
      DO 155 K=I,IPN
      IF (IABS(DAT(J,K+1)-DAT(J,K)) .GT. 10) GO TO 1000
      IF (IABS(DAT(J,K+1)-DATJI) .GT. 50) GO TO 1000
      Z=Z+DAT(J,K+1)
155   CONTINUE
      YL(J,1)=Z/10.
      SPT(J,1)=IK+I
1000  CONTINUE
500   CONTINUE
      IK=IK+6144
      GO TO 95
3000  CONTINUE
      WRITE (17,583) YL
583   FORMAT (3I7)
      WRITE (17,502) SF
502   FORMAT (F12.4)
      CALL PEFILE (1,1,126,IPT,'PCGSFS')
      CALL GETFIL ('PCGSFS')
      LIMP(11)=YL(1,2)
      LIMP(12)=YL(2,2)
      LIMP(13)=YL(3,2)
      DAD(2)=0
```

```
      CALL CIO (27,6,PO,2,IST,128,LIMP,DAD)
85    IF (IST .EQ. 0) GO TO 85
      IK=0
805   CALL CIO (22,6,PO,0,IST,3072,BAT,DAD)
905   IF (IST .EQ. 0) GO TO 905
      CALL BUFFER (BAT(1,1,2),BAT(3,1000,2))
      CALL CDRIVE
      DO 620 I=1,3
      CALL ERASE
      IP=2
      DO 610 J=1,1024
      IF (BAT(1,J,1) .EQ. -9999) GO TO 615
      CALL PLOT1 (IP,J-1,BAT(I,J,1)*3/8+512)
      DO 6151 L=1,2
      IF (SPT(I,L) .NE. IK+J) GO TO 6151
      CALL PLOT3 (1,20,-20)
      CALL PLOT3 (1,-40,0)
      CALL PLOT3 (1,20,20)
6151  CONTINUE
      IP=1
610   CONTINUE
      J=1
/I
615   SMM   3
/F
620   CONTINUE
      IF (BAT(1,J,1) .EQ. -9999) GO TO 621
      DAD(2)=DAD(2)+24
      IK=IK+1024
      GO TO 805
621   CONTINUE
/I
      EQP   22
/F
      END
      EOF
      EOF
```

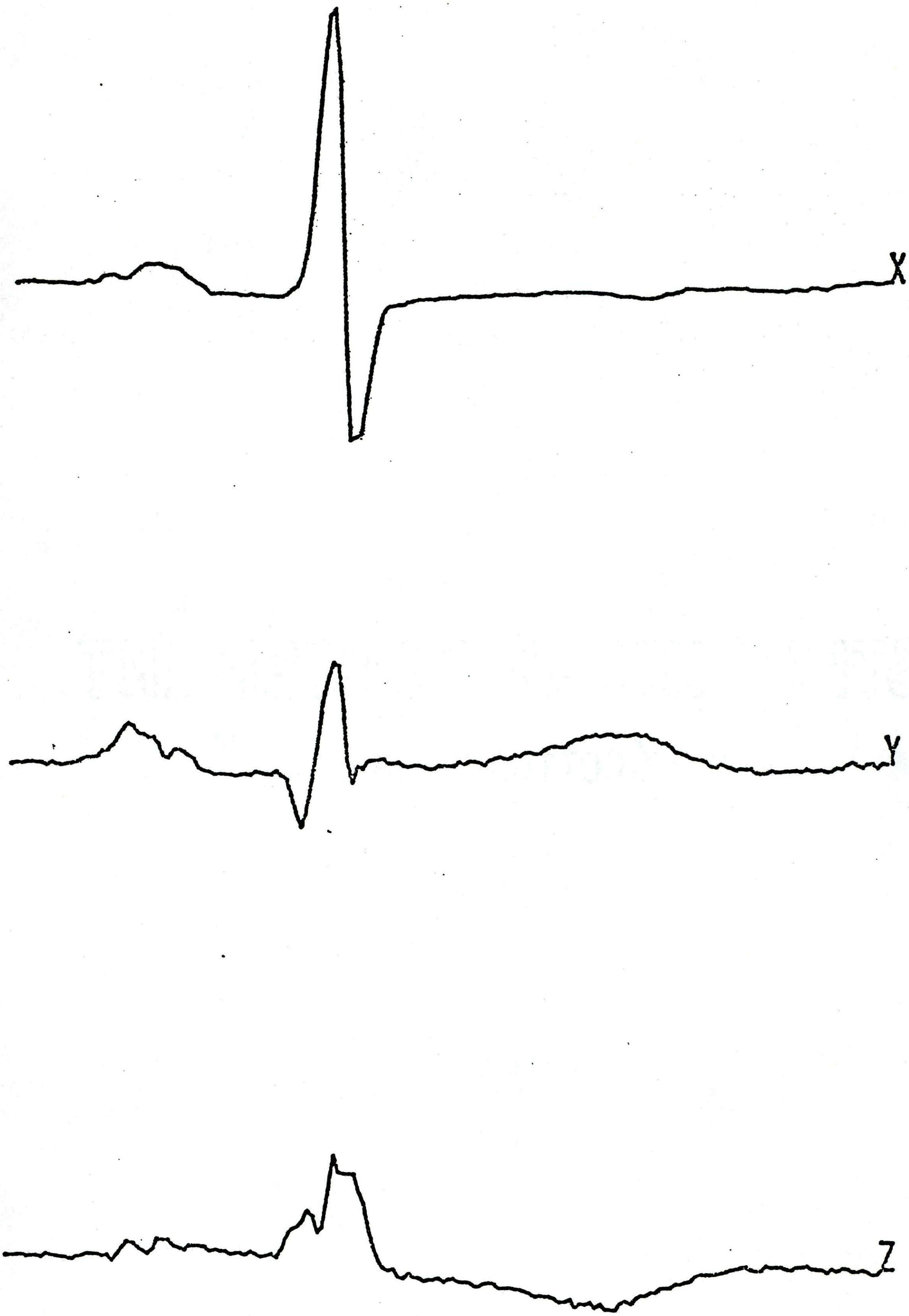
```
      DIMENSION NAME(10),MAG(9984),MXYZ(3,750)
      DIMENSION FIX(3)
      EQUIVALENCE (MAG,MXYZ)
      CALL PEFILE (1,1,126*30,IPT,`MEDIAN')
      CALL FILE (1,1)
      READ (4) ILIM,BND,EDD,CLOMP,MEDRR,           (MAG(I),I=1,ILIM)
      DO 10 I=1,3
      READ (4) ILIM,FIX,(MXYZ(I,J),J=1,ILIM)
      DO 10 J=1,ILIM
10    MXYZ(I,J)=MXYZ(I,J)+5000
      WRITE (39,20) ((MXYZ(I,J),I=1,3),J=1,ILIM,4)
20    FORMAT (3I4)
      WRITE (39,30)
30    FORMAT (`0')
```

EX. 5 EMR 6130 FORTRAN PROGRAM
TRANSFERS DATA FROM THE 6130 TO MIIS



Ex. 6 Median Wave

Produced by the 6130 (Fortran)



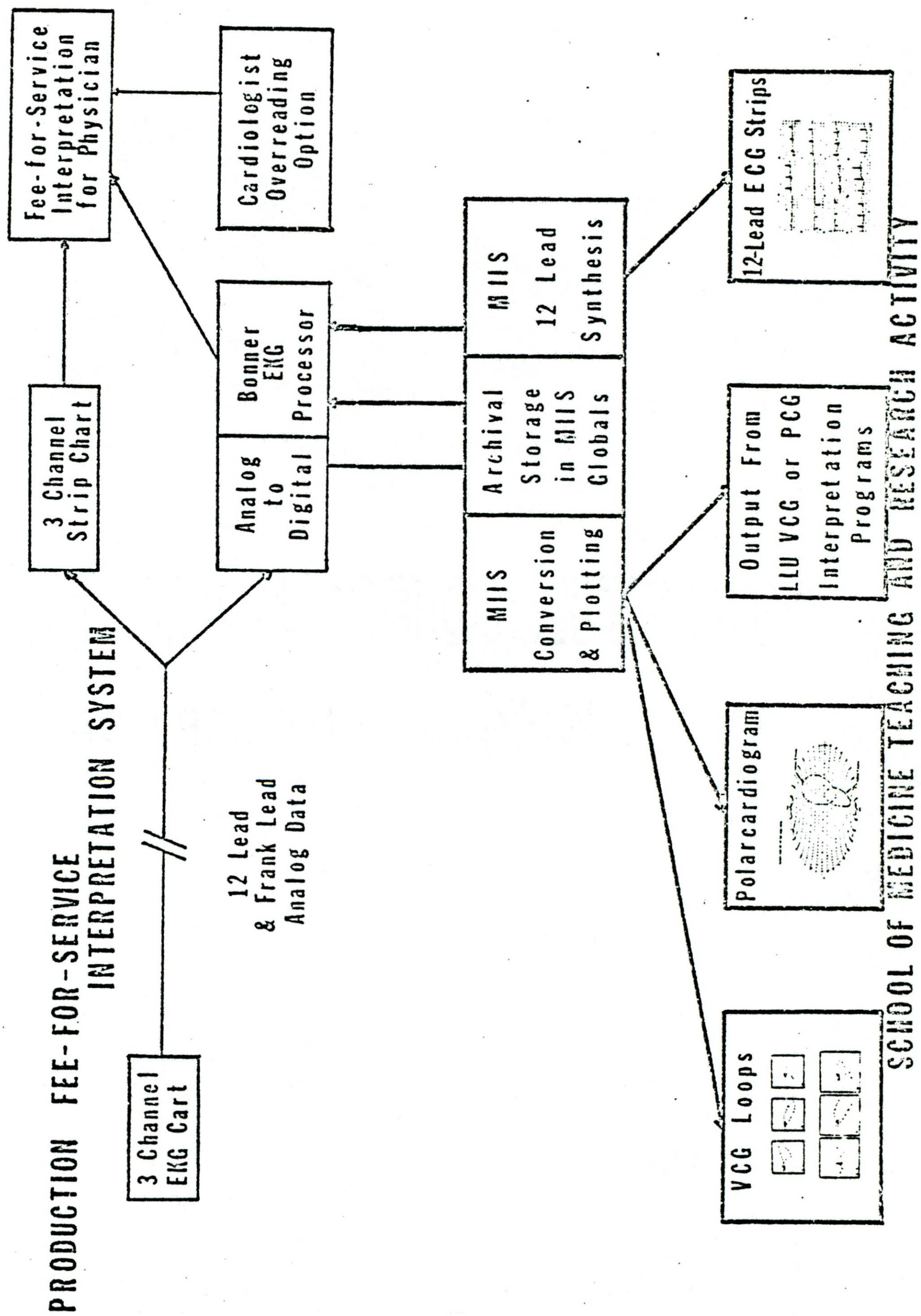
Ex. 7 Median Wave

Produced by the Tektronix (Basic)

```
100 FIND 2
110 DIM X(200),Y(200),Z(200)
120 I=0
130 I=I+1
140 INPUT @33:X(I),Y(I),Z(I)
150 IF X(I)=0 THEN 200
160 X(I)=X(I)-5000
170 Y(I)=Y(I)-5000
180 Z(I)=Z(I)-5000
190 GO TO 130
200 PAGE
210 WINDOW 0,I,-200,400
220 VIEWPORT 15,115,65,95
230 MOVE 1,X(1)
240 MOVE @1:1,X(1)
250 FOR K=1 TO I-1
260 DRAW K,X(K)
270 DRAW @1:K,X(K)
280 NEXT K
290 PRINT "X"
300 PRINT @1:"X"
310 VIEWPORT 15,115,35,65
320 MOVE 1,Y(1)
330 MOVE @1:1,Y(1)
340 FOR K=1 TO I-1
350 DRAW K,Y(K)
360 DRAW @1:K,Y(K)
370 NEXT K
380 PRINT "Y"
390 PRINT @1:"Y"
400 VIEWPORT 15,115,5,35
410 MOVE 1,Z(1)
420 MOVE @1:1,Z(1)
430 FOR K=1 TO I-1
440 DRAW K,Z(K)
450 DRAW @1:K,Z(K)
460 NEXT K
470 PRINT "Z"
480 PRINT @1:"Z"
490 END
```

Ex. 7 Median Wave Program

Produced by the Tektronix (Basic)



EX. 8
DIAGRAM OF OUTLINE OF THESIS

APPENDIX II

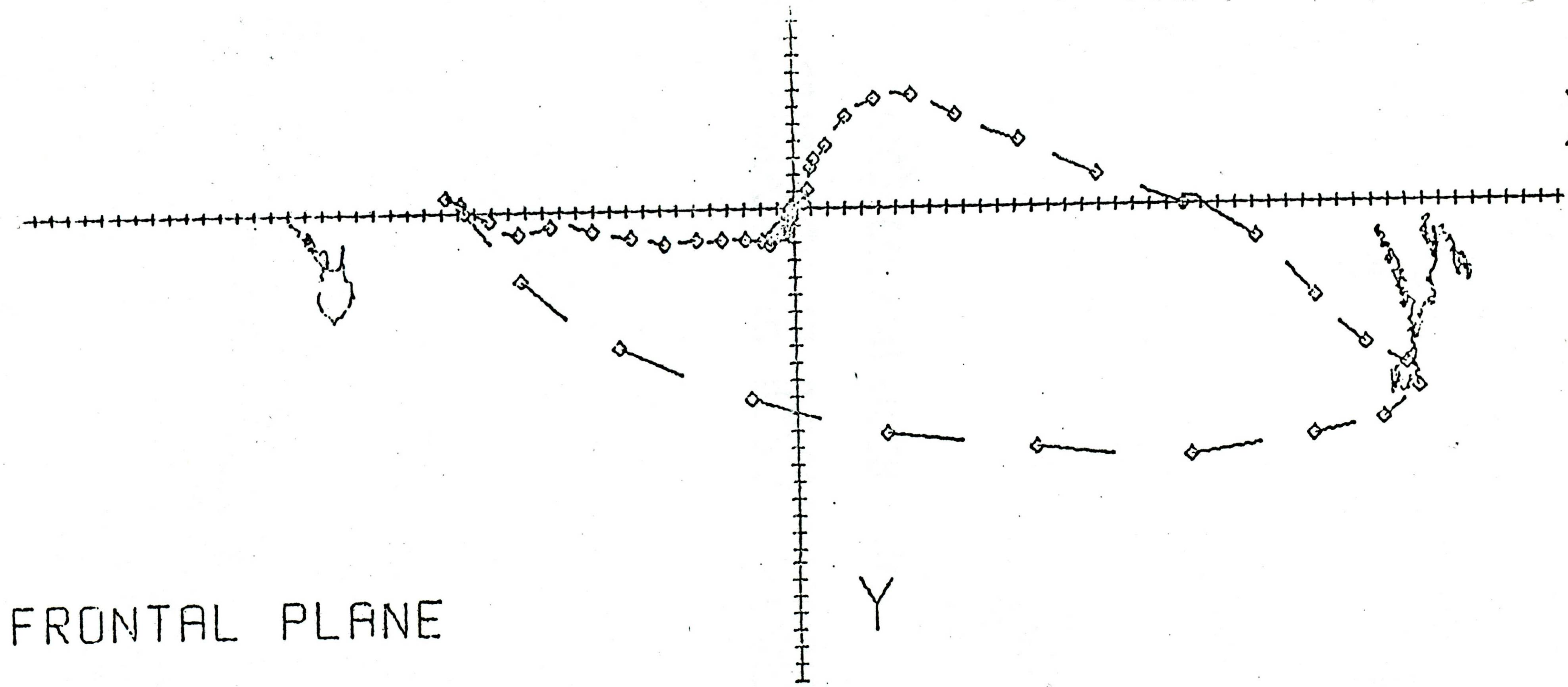
```
      INTEGER T(9984)
      INTEGER BND,EDD,CLOMP
      INTEGER MXYZ(750,3)
      REAL JIS,JUS
      DIMENSION FIX(3)
      CALL PEFILE (1,1,126*30,IPT,@MEDIAN@)
      CALL FILE (1,1)
      READ (4) ILIM,BND,EDD,CLOMP,MEDRR,(T(I),I=1,ILIM)
      DO 102 I=1,3
102   READ (4) ILIM,FIX,(MXYZ(J,I),J=1,ILIM)
      WRITE (17,17) ILIM
17    FORMAT (@ TIME SPAN = @,I3,@ MS.@)
      KKK=0
      DO 212 I=1,3
      DO 212 J=2,ILIM
      KKK=MAX0(MXYZ(J,I),KKK)
212   CONTINUE
      JUS=1
      IF (KKK.LT.200) JUS=2
      IF (KKK.LT.100) JUS=4
      IF (KKK.GT.400) JUS=.5
      CALL BUFFER (T,T(9984))
      CALL ERASE
      CALL CDRIVE
      VOOM=.75
      CALL PLOT2 (2,30,150)
      WRITE (8,37)
37    FORMAT ('3X')
      CALL PLOT2 (2,30,450)
      WRITE (8,38)
38    FORMAT ('3Y')
      CALL PLOT2 (2,30,750)
      WRITE (8,39)
39    FORMAT ('3Z')
      DO 747 L=1,3
      CALL PLOT1 (2,0,300*(L-1))
      CALL DRAW1 (0.,1000.,10.,10.,-VOOM,VOOM,3.,10.)
      CALL PUP
      DO 742 J=1,ILIM
742   CALL DRAW8 (1,FLOAT(J),MXYZ(J,L)*FIX(L))
747   CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
```

EX.1 6130 FORTRAN PROGRAM
PRODUCES VECTOR LOOPS

```
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,600,30)
      IF (JUS.EQ..5) WRITE(8,524)
      IF (JUS.EQ.1.) WRITE(8,521)
      IF (JUS.EQ.2.) WRITE(8,522)
      IF (JUS.EQ.4.) WRITE(8,523)
524   FORMAT ('1SCALE 1:.5')
521   FORMAT ('1SCALE 1:1')
522   FORMAT ('1SCALE 1:2')
523   FORMAT ('1SCALE 1:4')
      CALL PLOT2 (2,0,0)
      CALL DRAW4 (-500.,500.,10.,10.,-500.,500.,10.,10.)
      CALL PUP
      CALL PLOT2 (2,30,30)
      WRITE (8,41)
41    FORMAT ('2FRONTAL PLANE')
      CALL PLOT2 (2,950,530)
      WRITE (8,42)
      CALL PLOT2 (2,530,30)
      WRITE (8,43)
42    FORMAT ('3X')
43    FORMAT ('3Y')
      JIS=JUS
      DO 80 J=1,ILIM
      IF (J.EQ.331) JIS=2*JUS
      BA=MXYZ(J,1)*JIS
      BB=-MXYZ (J,2)*JIS
      ISYM=050
      IF (2*(J/2).EQ.J)  ISYM=1
      IF (J.LT.180) ISYM=1
      IF (J.GT.330) ISYM=1
      IF (J.EQ.180) CALL PUP
      IF (J.EQ.331) CALL PUP
      IF (J.EQ.329) ISYM=1
      IF (J.LT.180) BA=BA-300.
      IF (J.GT.330) BA=BA+350.
      CALL DRAW8 (ISYM,BA,BB)
80    CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,600,30)
      IF (JUS.EQ..5) WRITE(8,524)
      IF (JUS.EQ.1.) WRITE(8,521)
```

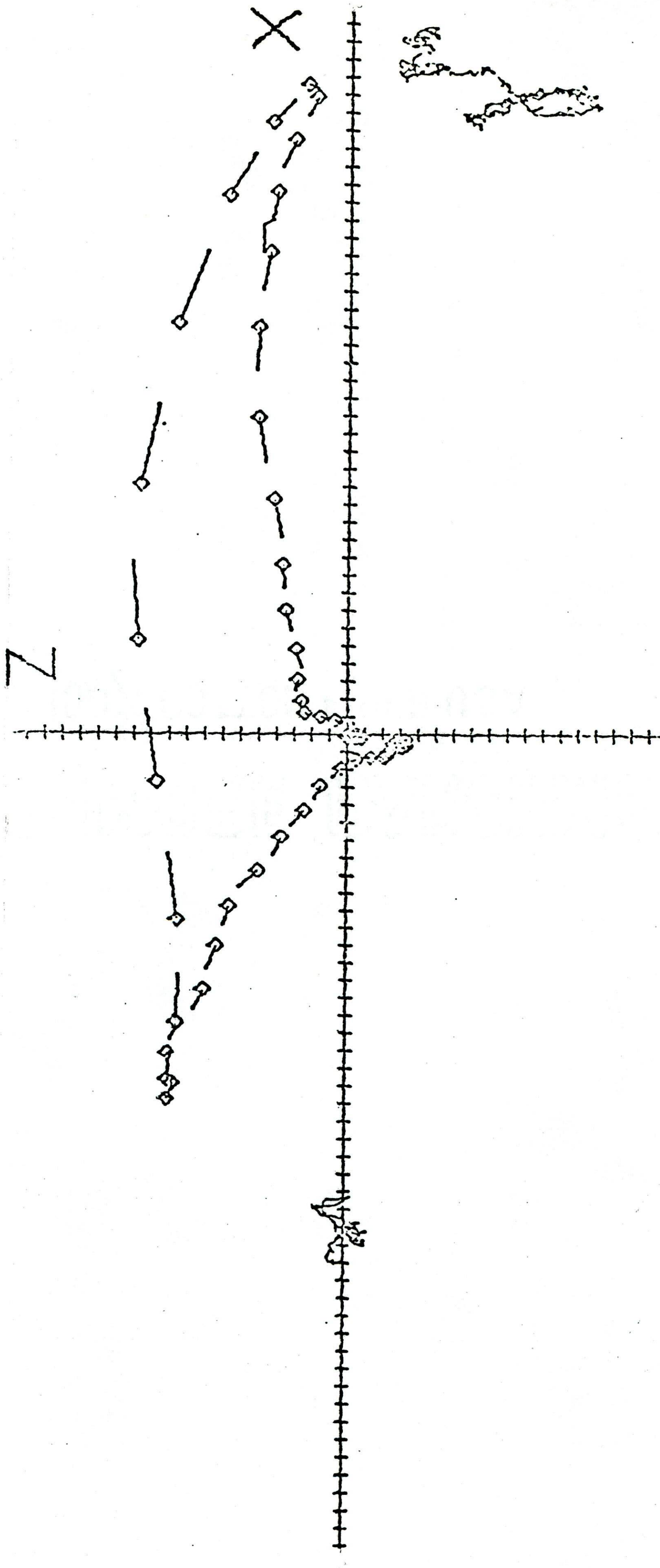
```
      IF (JUS.EQ.2.) WRITE(8,522)
      IF (JUS.EQ.4.) WRITE(8,523)
      CALL PLOT2 (2,0,0)
      CALL DRAW4 (-500.,500.,10.,10.,-500.,500.,10.,10.)
      CALL PUP
      CALL PLOT2 (2,30,30)
      WRITE (8,51)
51    FORMAT ('2TRANSVERSE PLANE')
      CALL PLOT2 (2,950,530)
      WRITE (8,42)
      CALL PLOT2 (2,530,960)
      WRITE (8,53)
53    FORMAT ('3Z')
      JIS=JUS
      DO 90 J=1,ILIM
      IF (J.EQ.331) JIS=2*JUS
      BA=MXYZ(J,1)*JIS
      BB=MXYZ(J,3)*JIS
      ISYM=050
      IF (2*(J/2).EQ.J)  ISYM=1
      IF (J.LT.180) ISYM=1
      IF (J.GT.330) ISYM=1
      IF (J.EQ.180) CALL PUP
      IF (J.EQ.331) CALL PUP
      IF (J.EQ.329) ISYM=1
      IF (J.LT.180) BA=BA-300.
      IF (J.GT.330) BA=BA+350.
      CALL DRAW8 (ISYM,BA,BB)
90    CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,600,30)
      IF (JUS.EQ..5) WRITE(8,524)
      IF (JUS.EQ.1.) WRITE(8,521)
      IF (JUS.EQ.2.) WRITE(8,522)
      IF (JUS.EQ.4.) WRITE(8,523)
      CALL PLOT2 (2,0,0)
      CALL DRAW4 (-500.,500.,10.,10.,-500.,500.,10.,10.)
      CALL PUP
      CALL PLOT2 (2,30,30)
      WRITE (8,61)
61    FORMAT ('2SAGITTAL PLANE')
      CALL PLOT2 (2,950,530)
      WRITE (8,53)
      CALL PLOT2 (2,530,960)
```

```
      WRITE (8,43)
      JIS=JUS
      DO 93 J=1,ILIM
      IF (J.EQ.331) JIS=2*JUS
      BA=MXYZ(J,3)*JIS
      BB=MXYZ(J,2)*JIS
      ISYM=050
      IF (2*(J/2).EQ.J)  ISYM=1
      IF (J.LT.180) ISYM=1
      IF (J.GT.330) ISYM=1
      IF (J.EQ.180) CALL PUP
      IF (J.EQ.331) CALL PUP
      IF (J.EQ.329) ISYM=1
      IF (J.LT.180) BA=BA-300.
      IF (J.GT.330) BA=BA+350.
      CALL DRAW8 (ISYM,BA,BB)
93    CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      END
      EOF
```

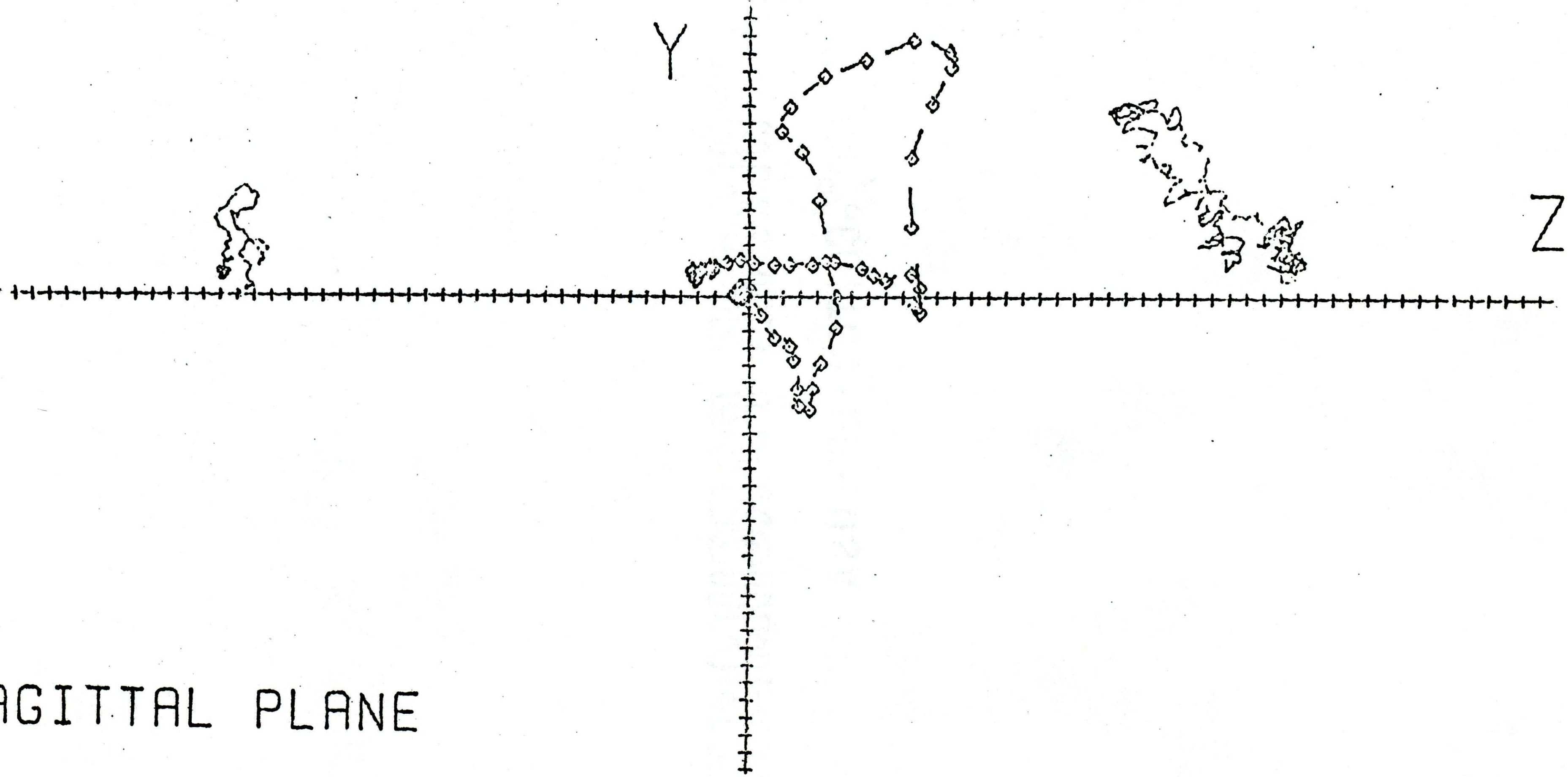


Ex. 2 Vector Loops

Produced by the 6130 (Fortran)



TRANSVERSE PLANE



SAGITTAL PLANE

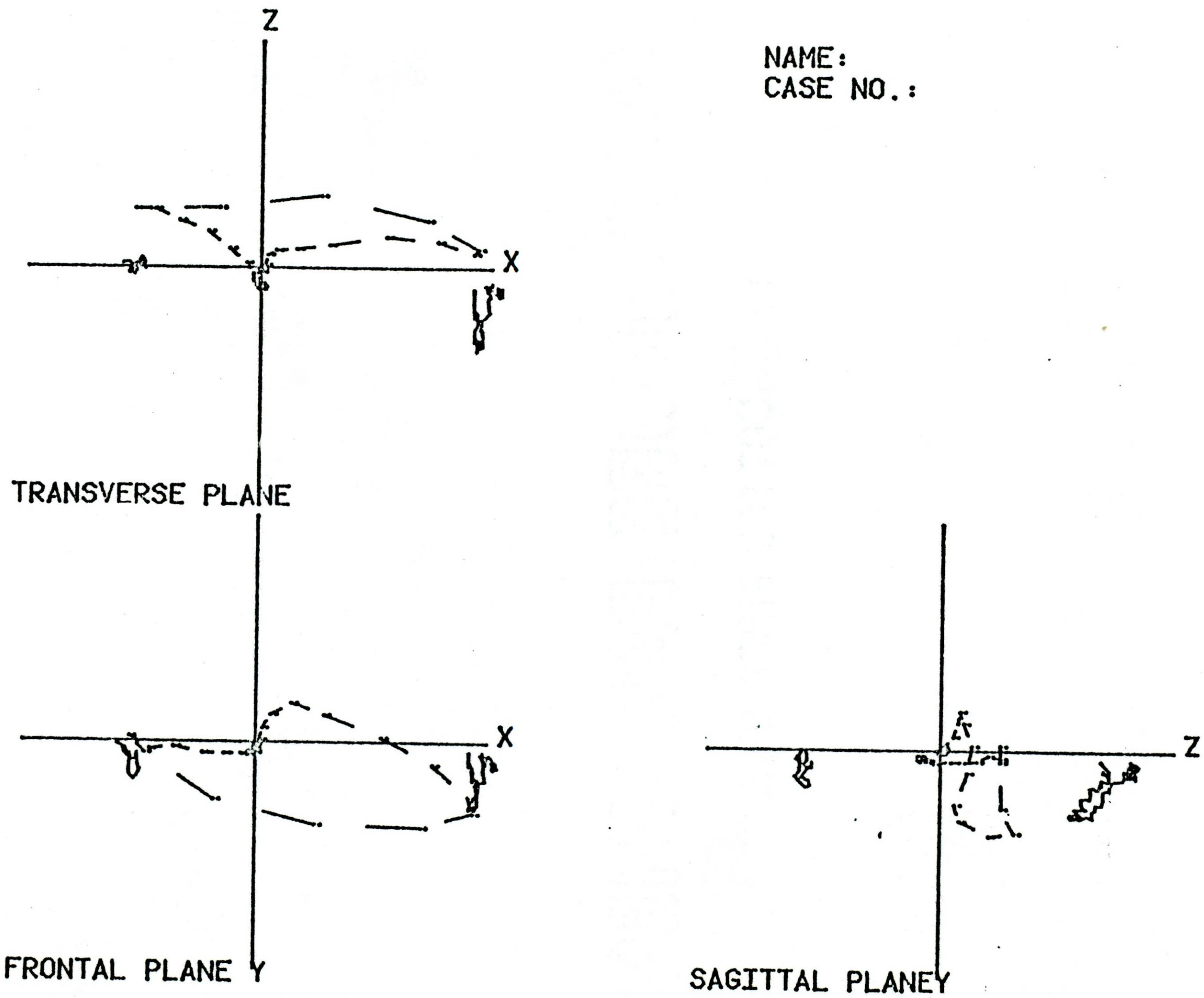
```
100 DIM A$(10)
110 FIND 2
120 DIM X(200),Y(200),Z(200)
130 I=0
140 I=I+1
150 INPUT @33:X(I),Y(I),Z(I)
160 IF X(I)=0 THEN 210
170 X(I)=X(I)-5000
180 Y(I)=Y(I)-5000
190 Z(I)=Z(I)-5000
200 GO TO 140
210 PAGE
220 Z3=I-1
230 WINDOW -400,400,-400,400
240 VIEWPORT 10,60,5,50
250 MOVE -380,0
260 MOVE @1:-380,0
270 DRAW 380,0
280 DRAW @1:380,0
290 MOVE 0,380
300 MOVE @1:0,380
310 DRAW 0,-380
320 DRAW @1:0,-380
330 MOVE X(1)-225,-Y(1)
340 MOVE @1:X(1)-225,-Y(1)
350 FOR I=1 TO Z3
360 D=X(I)
370 E=Y(I)
380 B=X(I)
390 C=-Y(I)
400 IF I<>41 THEN 430
410 MOVE X(I),-Y(I)
420 MOVE @1:X(I),-Y(I)
430 IF I<>90 THEN 460
440 MOVE X(I)+350,-Y(I)
450 MOVE @1:X(I)+350,-Y(I)
460 IF I>40 THEN 490
470 D=X(I)-225
480 GO TO 610
490 IF I<90 THEN 530
500 D=2*X(I)+350
510 E=2*Y(I)
520 GO TO 610
530 D=(X(I)+X(I+1))/2
540 E=(Y(I)+Y(I+1))/2
550 MOVE B,C
560 MOVE @1:B,C
570 PRINT "."
580 PRINT @1:"."
```

EX. 3 TEKTRONIX BASIC PROGRAM
PRODUCES VECTOR LOOPS

```
590 MOVE B,C
600 MOVE @1:B,C
610 DRAW D,-E
620 DRAW @1:D,-E
630 NEXT I
640 MOVE -400,-400
650 MOVE @1:-400,-400
660 PRINT "FRONTAL PLANE"
670 PRINT @1:"FRONTAL PLANE"
680 MOVE 0,-400
690 MOVE @1:0,-400
700 PRINT "Y"
710 PRINT @1:"Y"
720 MOVE 400,0
730 MOVE @1:400,0
740 PRINT "X"
750 PRINT @1:"X"
760 VIEWPORT 10,60,50,95
770 MOVE -380,0
780 MOVE @1:-380,0
790 DRAW 380,0
800 DRAW @1:380,0
810 MOVE 0,380
820 MOVE @1:0,380
830 DRAW 0,-380
840 DRAW @1:0,-380
850 MOVE X(1)-225,Z(1)
860 MOVE @1:X(1)-225,Z(1)
870 FOR I=1 TO Z3
880 D=X(I)
890 F=Z(I)
900 C=Z(I)
910 B=X(I)
920 IF I<>41 THEN 950
930 MOVE X(I),Z(I)
940 MOVE @1:X(I),Z(I)
950 IF I<>90 THEN 980
960 MOVE X(I)+350,Z(I)
970 MOVE @1:X(I)+350,Z(I)
980 IF I>40 THEN 1010
990 D=X(I)-225
1000 GO TO 1130
1010 IF I<90 THEN 1050
1020 D=2*X(I)+350
1030 F=2*Z(I)
1040 GO TO 1130
1050 D=(X(I)+X(I+1))/2
```

```
1060 F=(Z(I)+Z(I+1))/2
1070 MOVE B,C
1080 MOVE @1:B,C
1090 PRINT "."
1100 PRINT @1:"."
1110 MOVE B,C
1120 MOVE @1:B,C
1130 DRAW D,F
1140 DRAW @1:D,F
1150 NEXT I
1160 MOVE -400,-400
1170 MOVE @1:-400,-400
1180 PRINT "TRANSVERSE PLANE"
1190 PRINT @1:"TRANSVERSE PLANE"
1200 MOVE 0,400
1210 MOVE @1:0,400
1220 PRINT "Z"
1230 PRINT @1:"Z"
1240 MOVE 400,0
1250 MOVE @1:400,0
1260 PRINT "X"
1270 PRINT @1:"X"
1280 VIEWPORT 80,130,5,50
1290 MOVE -380,0
1300 MOVE @1:-380,0
1310 DRAW 380,0
1320 DRAW @1:380,0
1330 MOVE 0,380
1340 MOVE @1:0,380
1350 DRAW 0,-380
1360 DRAW @1:0,-380
1370 MOVE Z(I)-225,-Y(I)
1380 MOVE @1:Z(I)-225,-Y(I)
1390 FOR I=1 TO Z3
1400 D=Z(I)
1410 E=Y(I)
1420 B=Z(I)
1430 C=-Y(I)
1440 IF I<>41 THEN 1470
1450 MOVE Z(I),-Y(I)
1460 MOVE @1:Z(I),-Y(I)
1470 IF I<>90 THEN 1500
1480 MOVE Z(I)+300,-Y(I)
1490 MOVE @1:Z(I)+300,-Y(I)
1500 IF I>40 THEN 1530
1510 D=Z(I)-225
1520 GO TO 1650
1530 IF I<90 THEN 1570
1540 D=2*Z(I)+350
1550 E=2*Y(I)
```

```
1560 GO TO 1650
1570 D=(Z(I)+Z(I+1))/2
1580 E=(Y(I)+Y(I+1))/2
1590 MOVE B,C
1600 MOVE @1:B,C
1610 PRINT "."
1620 PRINT @1:"."
1630 MOVE B,C
1640 MOVE @1:B,C
1650 DRAW D,-E
1660 DRAW @1:D,-E
1670 NEXT I
1680 MOVE -400,-400
1690 MOVE @1:-400,-400
1700 PRINT "SAGITTAL PLANE"
1710 PRINT @1:"SAGITTAL PLANE"
1720 MOVE 0,-400
1730 MOVE @1:0,-400
1740 PRINT "Y"
1750 PRINT @1:"Y"
1760 MOVE 400,0
1770 MOVE @1:400,0
1780 PRINT "Z"
1790 PRINT @1:"Z"
1800 VIEWPORT 80,130,50,95
1810 MOVE -300,350
1820 MOVE @1:-300,350
1830 PRINT "NAME:"
1840 PRINT @1:"NAME:"
1850 MOVE -300,300
1860 MOVE @1:-300,300
1870 PRINT "CASE NO.:"
1880 PRINT @1:"CASE NO.:"
1890 END
```



EX. 4 VECTOR LOOPS
PRODUCED BY THE TEKTRONIX (BASIC)

APPENDIX III

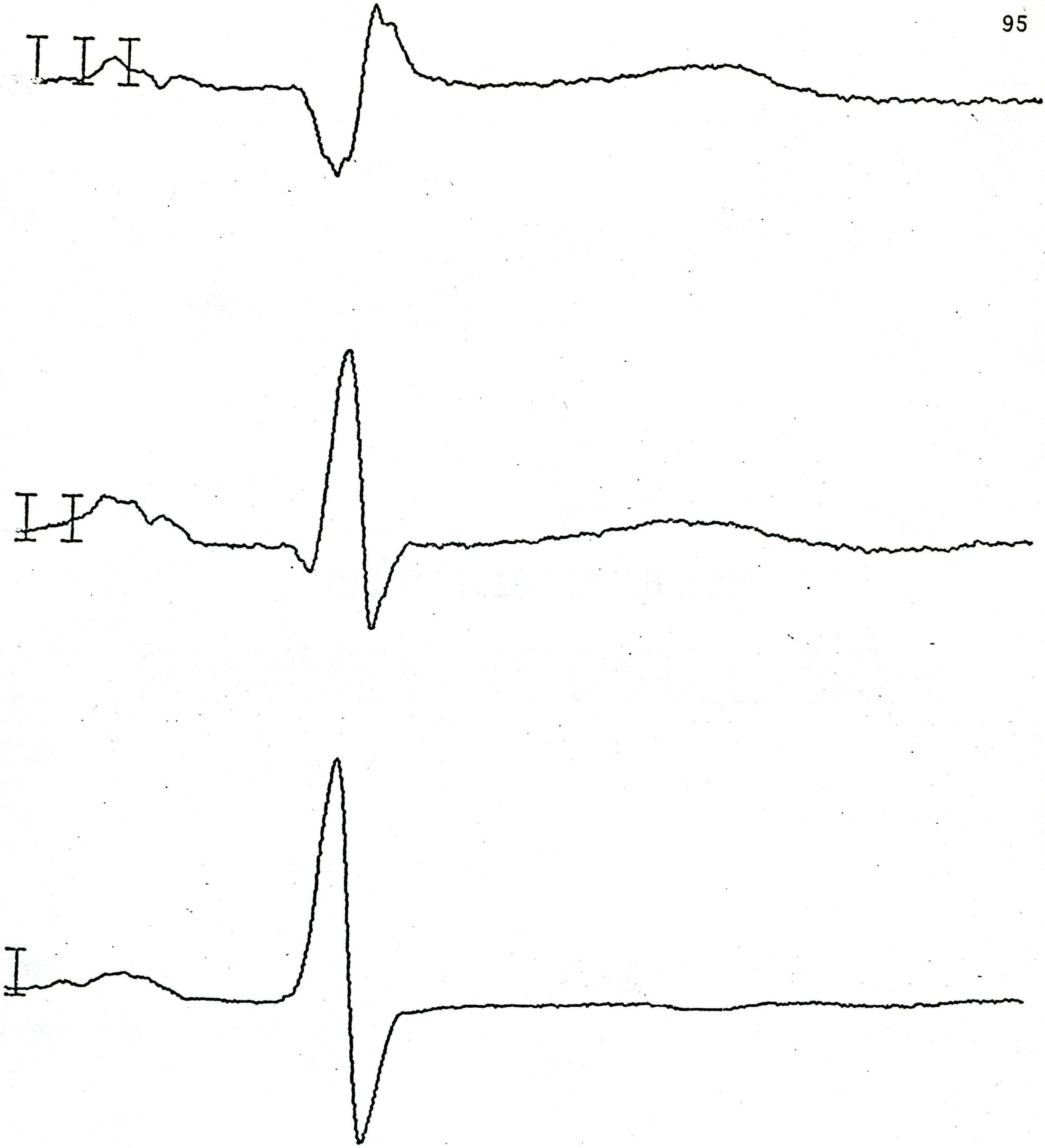
```
      INTEGER T(9984)
      INTEGER BND,EDD,CLOMP
      INTEGER MXYZ(750,3)
      REAL JIS,JUS
      DIMENSION FIX(3)
      CALL PEFILE (1,1,126*30,IPT,@MEDIAN@)
      CALL FILE (1,1)
      READ (4) ILIM,BND,EDD,CLOMP,MEDRR,(T(I),I=1,ILIM)
      DO 102 I=1,3
102   READ (4) ILIM,FIX,(MXYZ(J,I),J=1,ILIM)
      CALL BUFFER (T,T(9984))
      CALL ERASE
      CALL CDRIVE
      VOOM=.75
      CALL PLOT2 (2,30,150)
      WRITE (8,37)
37    FORMAT ('3X')
      CALL PLOT2 (2,30,450)
      WRITE (8,38)
38    FORMAT ('3Y')
      CALL PLOT2 (2,30,750)
      WRITE (8,39)
39    FORMAT ('3Z')
      DO 747 L=1,3
      CALL PLOT1 (2,0,300*(L-1))
      CALL DRAW1 (0.,1000.,10.,10.,-VOOM,VOOM,3.,10.)
      CALL PUP
      DO 742 J=1,ILIM
742   CALL DRAW8 (1,FLOAT(J),MXYZ(J,L)*FIX(L))
747   CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,30,150)
      WRITE (8,137)
137   FORMAT ('3I')
      CALL PLOT2 (2,30,450)
      WRITE (8,138)
138   FORMAT ('3II')
```

___ EX. 1 EMR 6130 FORTRAN PROGRAM ___
SYNTHESIZES 12 LEADS

```
      CALL PLOT2 (2,30,750)
      WRITE (8,139)
139   FORMAT ('3III')
      DO 391 L=1,3
      IF (L.EQ.1) PHI=( 3.14159/2)
      IF (L.EQ.2) PHI=(5*3.14159/6)
      IF (L.EQ.3) PHI=(-5*3.14159/6)
      CALL PLOT1 (2,0,300*(L-1))
      CALL DRAW1 (0.,1000.,10.,10.,-VOOM,VOOM,3.,10.)
      CALL PUP
      DO 392 J=1,ILIM
      BAA=MXYZ(J,1)
      BAB=-MXYZ(J,2)
      IF (MXYZ(J,1).EQ.0) GOTO 808
      IF (MXYZ(J,2).EQ.0) GOTO 909
      THET=ATAN2(BAB,BAA)
      GOTO 100
808   IF (MXYZ(J,2).LT.0) THET=(3.14159*.5)
      IF (MXYZ(J,2).GT.0) THET=(-3.14159*.5)
      GOTO 100
909   IF (MXYZ(J,1).GE.0) THET=0
      IF (MXYZ(J,1).LT.0) THET=3.14159
100   Y=SQRT(BAA**2+BAB**2)*SIN(THET+PHI)
392   CALL DRAW8 (1,FLOAT(J),Y*FIX(L))
391   CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,30,150)
      WRITE (8,237)
237   FORMAT ('3AVL')
      CALL PLOT2 (2,30,450)
      WRITE (8,238)
238   FORMAT ('3AVF')
      CALL PLOT2 (2,30,750)
      WRITE (8,239)
239   FORMAT ('3AVR')
      DO 491 L=1,3
      IF (L.EQ.1) PHI=(3.14159/3)
      IF (L.EQ.2) PHI=3.14159
      IF (L.EQ.3) PHI=(-3.14159/3)
      CALL PLOT1 (2,0,300*(L-1))
      CALL DRAW1 (0.,1000.,10.,10.,-VOOM,VOOM,3.,10.)
      CALL PUP
      DO 492 J=1,ILIM
      BAA=MXYZ(J,1)
      BAB=-MXYZ(J,2)
```

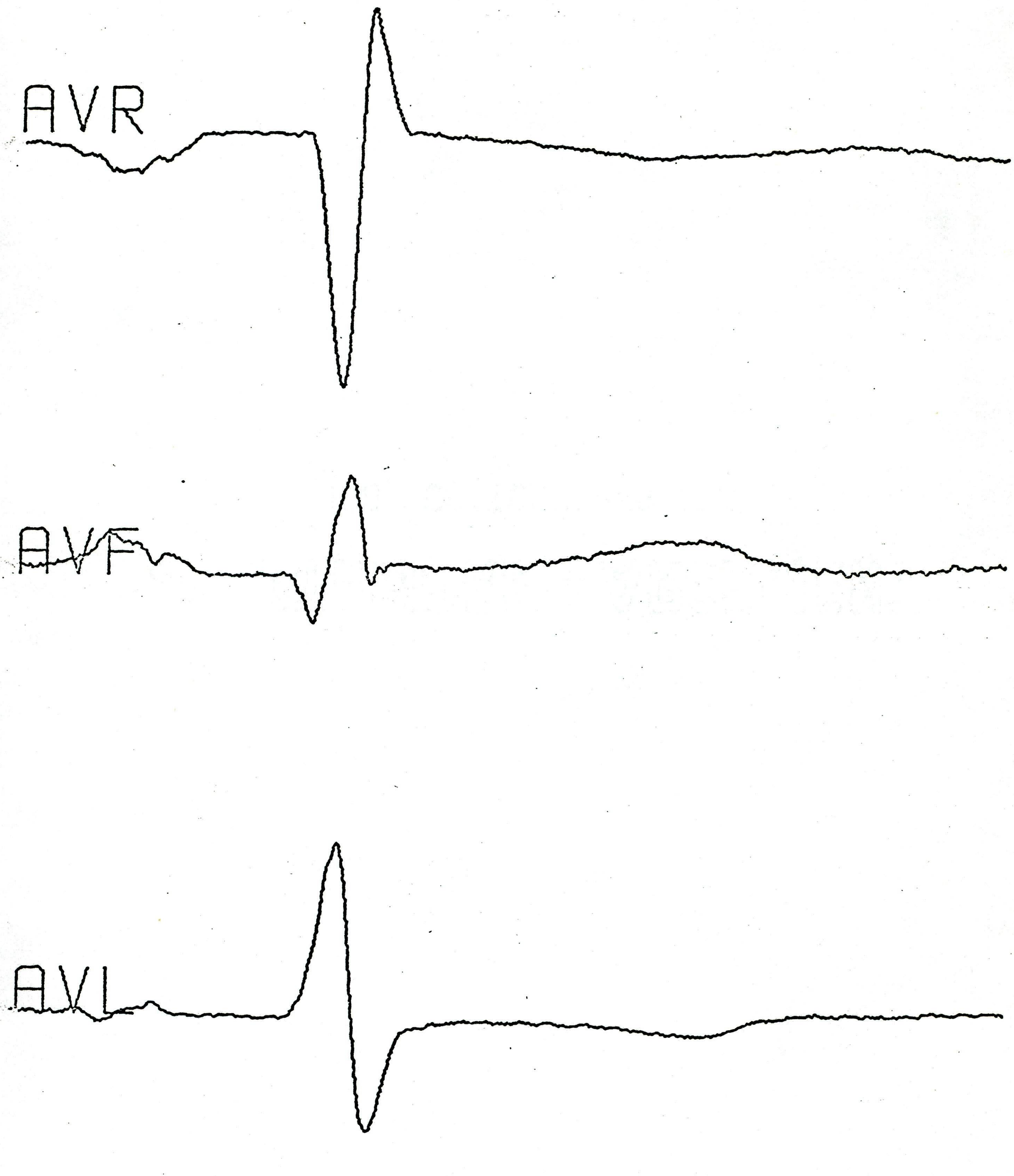
```
      IF (MXYZ(J,1).EQ.0) GOTO 809
      IF (MXYZ(J,2).EQ.0) GOTO 910
      THET=ATAN2(BAB,BAA)
      GO TO 101
809   IF (MXYZ(J,2).LT.0) THET=(3.14159*.5)
      IF (MXYZ(J,2).GT.0) THET=(-3.14159*.5)
      GO TO 101
910   IF (MXYZ(J,1).GE.0) THET=0
      IF (MXYZ(J,1).LT.0) THET=3.14159
101   Y=SQRT(BAA**2+BAB**2)*SIN(THET+PHI)
492   CALL DRAW8 (1,FLOAT(J),Y*FIX(L))
491   CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,30,150)
      WRITE (8,337)
337   FORMAT ('3V1')
      CALL PLOT2 (2,30,450)
      WRITE (8,338)
338   FORMAT ('3V2')
      CALL PLOT2 (2,30,750)
      WRITE (8,339)
339   FORMAT ('3V3')
      DO 591 L=1,3
      CALL PLOT1 (2,0,300*(L-1))
      CALL DRAW1 (0.,1000.,10.,10.,-VOOM,VOOM,3.,10.)
      CALL PUP
      IF (L.EQ.1) PHI=(17*3.14159/18)
      IF (L.EQ.2) PHI=(-17*3.14159/18)
      IF (L.EQ.3) PHI=(-15*3.14159/18)
      DO 592 J=1,ILIM
      BAA=MXYZ(J,1)
      BAC=MXYZ(J,3)
      IF (MXYZ(J,1).EQ.0) GOTO 810
      IF (MXYZ(J,3).EQ.0) GOTO 911
      THET=ATAN2(BAC,BAA)
      GO TO 103
810   IF (MXYZ(J,3).GT.0) THET=(3.14159*.5)
      IF (MXYZ(J,3).LT.0) THET=(-3.14159*.5)
      GO TO 103
911   IF (MXYZ(J,1).GE.0) THET=0
      IF (MXYZ(J,1).LT.0) THET=3.14159
103   Y=SQRT(BAA**2+BAC**2)*SIN(THET-PHI)
592   CALL DRAW8 (1,FLOAT(J),Y*FIX(L))
591   CONTINUE
      CALL HRDCPY (T,9984)
```

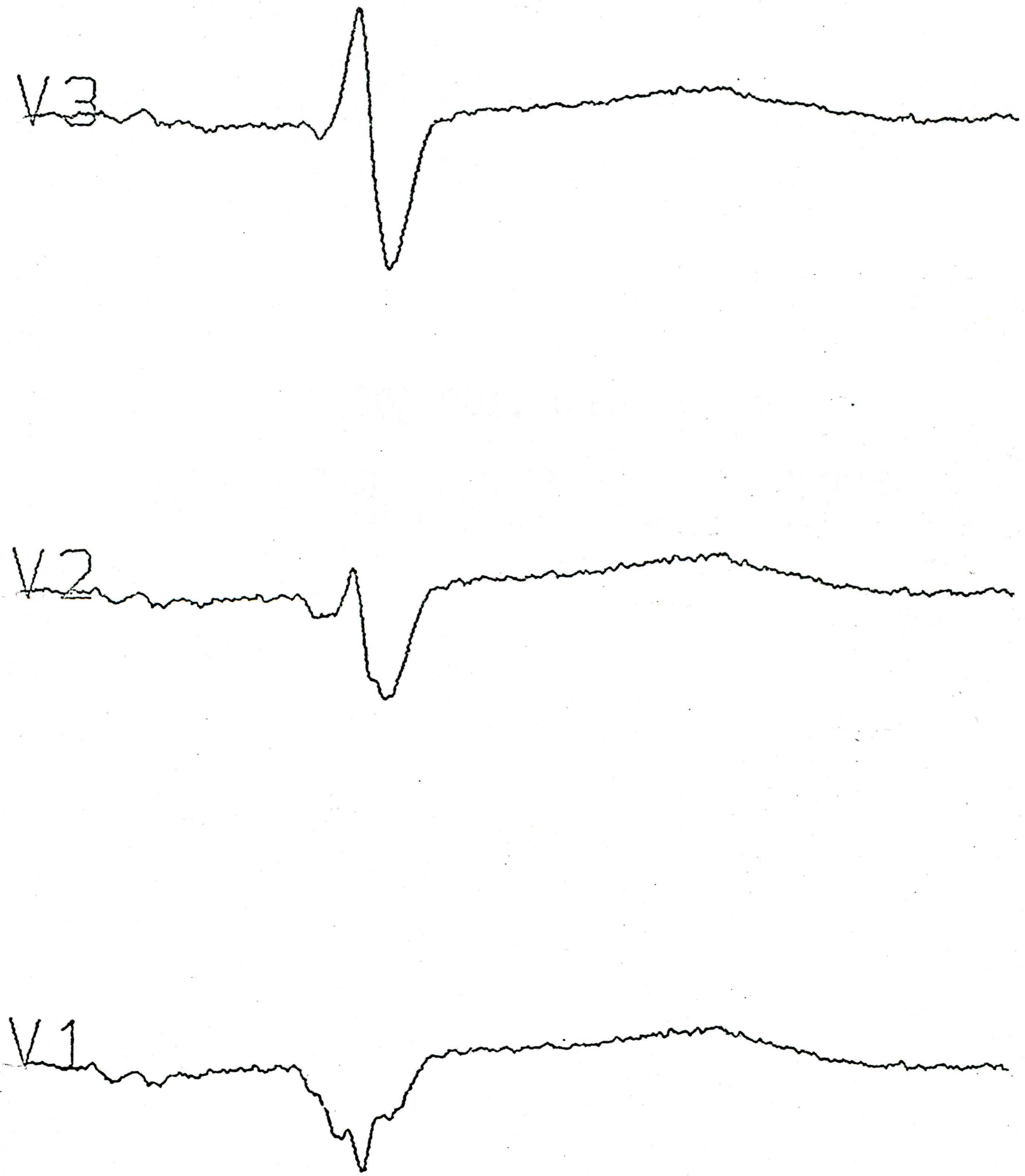
```
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT2 (2,30,150)
      WRITE (8,437)
437   FORMAT ('3V4')
      CALL PLOT2 (2,30,450)
      WRITE (8,438)
438   FORMAT ('3V5')
      CALL PLOT2 (2,30,750)
      WRITE (8,439)
439   FORMAT ('3V6')
      DO 691 L=1,3
      CALL PLOT1 (2,0,300*(L-1))
      CALL DRAW1 (0.,1000.,10.,10.,-VOOM,VOOM,3.,10.)
      CALL PUP
      IF (L.EQ.1) PHI=(-13*3.14159/18)
      IF (L.EQ.2) PHI=(-11*3.14159/18)
      IF (L.EQ.3) PHI=(-3.14159/2)
      DO 692 J=1,ILIM
      BAA=MXYZ(J,1)
      BAC=MXYZ(J,3)
      IF (MXYZ(J,1).EQ.0) GOTO 811
      IF (MXYZ(J,3).EQ.0) GOTO 912
      THET=ATAN2(BAC,BAA)
      GO TO 104
811   IF (MXYZ(J,3).GT.0) THET=(3.14159*.5)
      IF (MXYZ(J,3).LT.0) THET=(-3.14159*.5)
      GO TO 104
912   IF (MXYZ(J,1).GE.0) THET=0
      IF (MXYZ(J,1).LT.0) THET=3.14159
104   Y=SQRT(BAA**2+BAC**2)*SIN(THET-PHI)
692   CALL DRAW8 (1,FLOAT(J),Y*FIX(L))
691   CONTINUE
      CALL HRDCPY (T,9984)
      PAUSE
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      END
      EOF
```

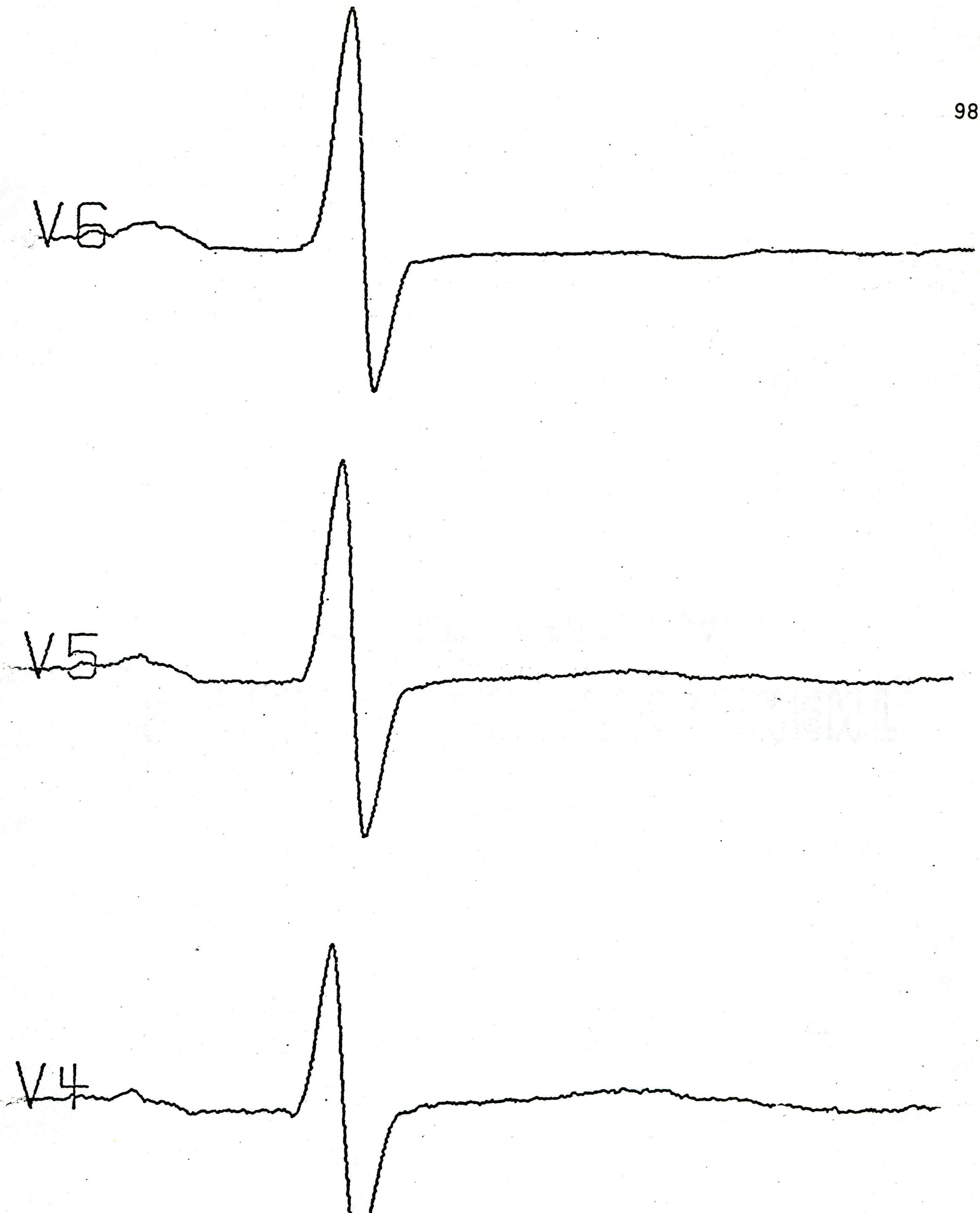


Ex. 2 12 Lead Synthesis

Produced by the 6130 (Fortran)


AVR
AVF
AVL


V3
V2
V1

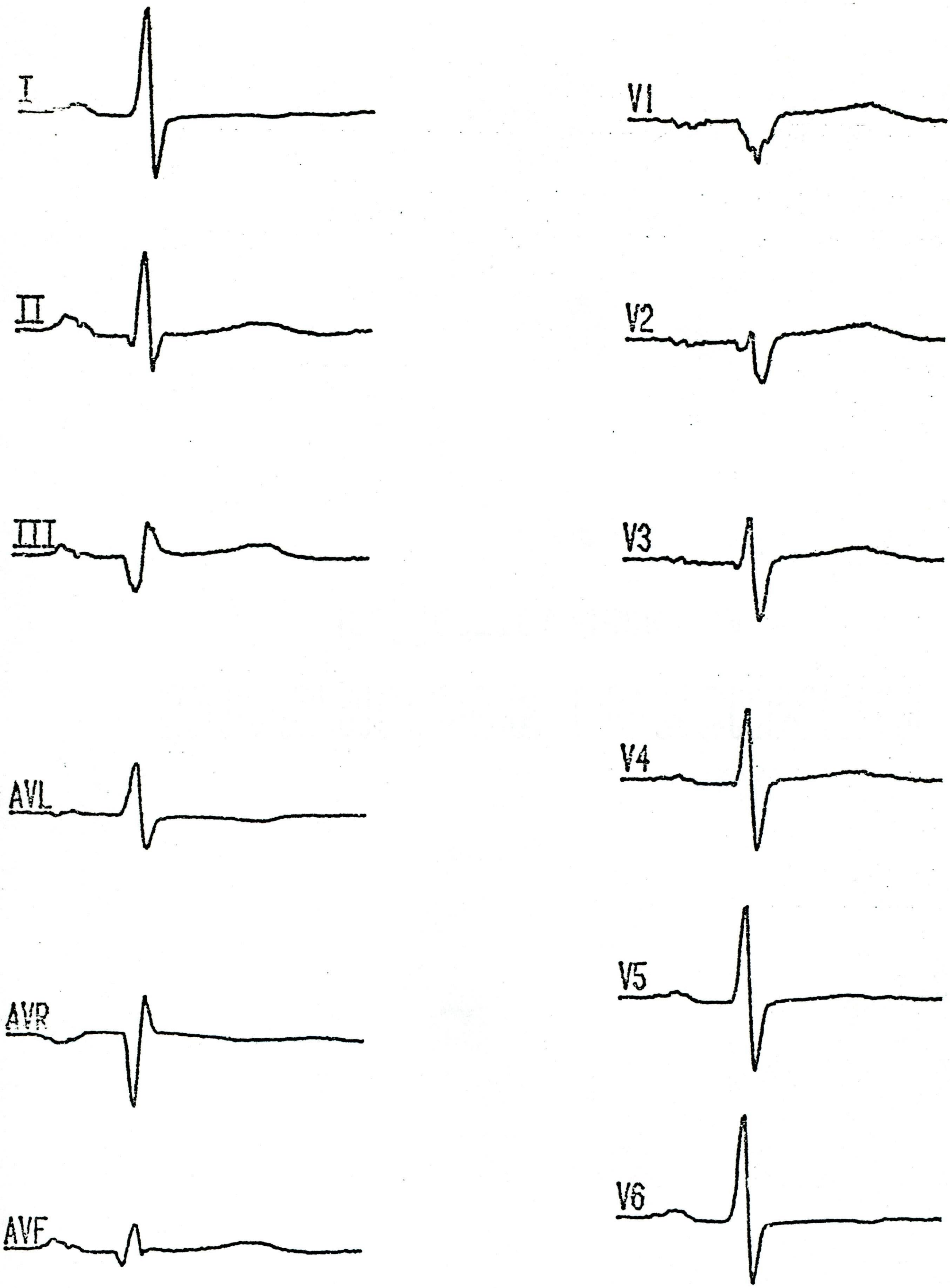

V6
V5
V4

```
100 DIM A$(10)
110 DIM X(200),Y(200),Z(200),P(3)
120 FIND 2
130 P(1)=-3.14159/2
140 P(2)=-5*3.14159/6
150 P(3)=5*3.14159/6
160 I=0
170 I=I+1
180 INPUT @33:X(I),Y(I),Z(I)
190 IF X(I)=0 THEN 240
200 X(I)=X(I)-5000
210 Y(I)=Y(I)-5000
220 Z(I)=Z(I)-5000
230 GO TO 170
240 PAGE
245 Z3=I-1
250 WINDOW 0,200,-370,370
260 FOR J=1 TO 3
270 VIEWPORT 5,55,95-J*15,95-(J-1)*15
280 FOR I=1 TO Z3
290 IF X(I)=0 THEN 340
300 T=ATN(-Y(I)/X(I))
310 IF X(I)=>0 THEN 350
320 T=T+3.14159
330 GO TO 350
340 T=-3.14159/2
350 S=SQR(X(I)^2+Y(I)^2)*SIN(T-P(J))
360 IF I<>1 THEN 520
370 MOVE I,S+30
380 MOVE @1:I,S+30
390 IF J=1 THEN 420
400 IF J=2 THEN 450
410 IF J=3 THEN 480
420 PRINT "I"
430 PRINT @1:"I"
440 GO TO 500
450 PRINT "II"
460 PRINT @1:"II"
470 GO TO 500
480 PRINT "III"
490 PRINT @1:"III"
500 MOVE I,S
510 MOVE @1:I,S
520 DRAW I,S
530 DRAW @1:I,S
540 NEXT I
550 NEXT J
```

-- EX. 3 TEKTRONIX BASIC PROGRAM --
SYNTHESIZES 12 LEADS

```
560 P(1)=-3.14159/3
570 P(2)=3.14159/3
580 P(3)=3.14159
590 FOR J=1 TO 3
600 VIEWPORT 5,55,50-J*15,45-(J-1)*15
610 FOR I=1 TO Z3
620 IF X(I)=0 THEN 670
630 T=ATN(-Y(I)/X(I))
640 IF X(I)=>0 THEN 680
650 T=T+3.14159
660 GO TO 680
670 T=-3.14159/2
680 S=SQR(X(I)^2+Y(I)^2)*SIN(T-P(J))
690 IF I<>1 THEN 850
700 MOVE I,S+30
710 MOVE @1:I,S+30
720 IF J=1 THEN 750
730 IF J=2 THEN 780
740 IF J=3 THEN 810
750 PRINT "AVL"
760 PRINT @1:"AVL"
770 GO TO 830
780 PRINT "AVR"
790 PRINT @1:"AVR"
800 GO TO 830
810 PRINT "AVF"
820 PRINT @1:"AVF"
830 MOVE I,S
840 MOVE @1:I,S
850 DRAW I,S
860 DRAW @1:I,S
870 NEXT I
880 NEXT J
890 P(1)=17*3.14159/18
900 P(2)=-17*3.14159/18
910 P(3)=-15*3.14159/18
920 FOR J=1 TO 3
930 VIEWPORT 80,130,95-J*15,95-(J-1)*15
940 FOR I=1 TO Z3
950 IF X(I)=0 THEN 1000
960 T=ATN(Z(I)/X(I))
970 IF X(I)=>0 THEN 1010
980 T=T+3.14159
990 GO TO 1010
1000 T=-3.14159/2
1010 S=SQR(X(I)^2+Z(I)^2)*SIN(T-P(J))
1020 IF I<>1 THEN 1180
1030 MOVE I,S+30
1040 MOVE @1:I,S+30
1050 IF J=1 THEN 1080
```

```
1060 IF J=2 THEN 1110
1070 IF J=3 THEN 1140
1080 PRINT "V1"
1090 PRINT @1:"V1"
1100 GO TO 1160
1110 PRINT "V2"
1120 PRINT @1:"V2"
1130 GO TO 1160
1140 PRINT "V3"
1150 PRINT @1:"V3"
1160 MOVE I,S
1170 MOVE @1:I,S
1180 DRAW I,S
1190 DRAW @1:I,S
1200 NEXT I
1210 NEXT J
1220 P(1)=-13*3.14159/18
1230 P(2)=-11*3.14159/18
1240 P(3)=-3.14159/2
1250 FOR J=1 TO 3
1260 VIEWPORT 80,130,50-J*15,50-(J-1)*15
1270 FOR I=1 TO Z3
1280 IF X(I)=0 THEN 1330
1290 T=ATN(Z(I)/X(I))
1300 IF X(I)=>0 THEN 1340
1310 T=T+3.14159
1320 GO TO 1340
1330 T=-3.14159/2
1340 S=SQR(X(I)^2+Z(I)^2)*SIN(T-P(J))
1350 IF I<>1 THEN 1510
1360 MOVE I,S+30
1370 MOVE @1:I,S+30
1380 IF J=1 THEN 1410
1390 IF J=2 THEN 1440
1400 IF J=3 THEN 1470
1410 PRINT "V4"
1420 PRINT @1:"V4"
1430 GO TO 1490
1440 PRINT "V5"
1450 PRINT @1:"V5"
1460 GO TO 1490
1470 PRINT "V6"
1480 PRINT @1:"V6"
1490 MOVE I,S
1500 MOVE @1:I,S
1510 DRAW I,S
1520 DRAW @1:I,S
1530 NEXT I
1540 NEXT J
1550 END
```



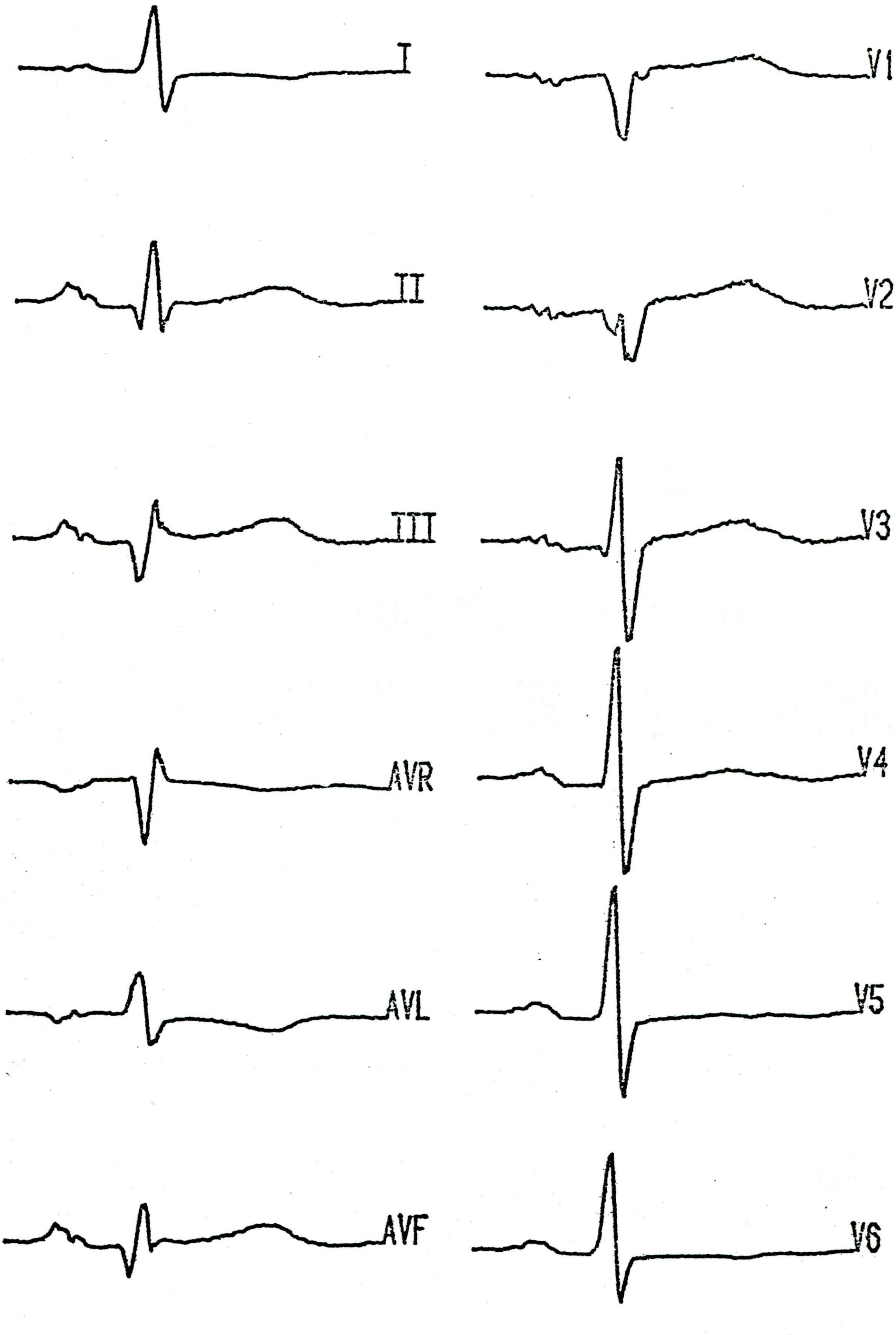
Ex. 4 12 Lead Synthesis

Produced by the Tektronix (Basic)

```
100 DIM A$(10)
110 FIND 2
120 DIM L(200),A(12),B(12),C(12)
130 DIM X(200),Y(200),Z(200)
140 I=0
150 I=I+1
160 INPUT @33:X(I),Y(I),Z(I)
170 IF X(I)=0 THEN 220
180 X(I)=X(I)-5000
190 Y(I)=Y(I)-5000
200 Z(I)=Z(I)-5000
210 GO TO 150
220 PAGE
230 A(1)=86
240 A(2)=32
250 A(3)=-54
260 A(4)=-59
270 A(5)=70
280 A(6)=-11
290 A(7)=-70
300 A(8)=6
310 A(9)=120
320 A(10)=165
330 A(11)=153
340 A(12)=113
350 B(1)=-32
360 B(2)=145
370 B(3)=177
380 B(4)=-56.5
390 B(5)=-104.5
400 B(6)=161
410 B(7)=21.33
420 B(8)=22.33
430 B(9)=13.33
440 B(10)=17.33
450 B(11)=17.33
460 B(12)=10.33
470 C(1)=8
480 C(2)=-18
490 C(3)=-26
500 C(4)=5
510 C(5)=17
520 C(6)=-22
530 C(7)=-124.67
540 C(8)=-188.67
550 C(9)=-173.67
560 C(10)=-81.67
570 C(11)=-11.67
580 C(12)=31.33
590 FOR K=1 TO 12
600 WINDOW 0,I,-40000,60000
```

-- Ex.5 Tektronix Basic Program --
Synthesis the 12 Leads (Dower Method)

```
610 IF K>6 THEN 640
620 VIEWPORT 15,60,95-K*15,110-K*15
630 GO TO 650
640 VIEWPORT 70,115,95-(K-6)*15,110-(K-6)*15
650 FOR J=1 TO I-1
660 L(J)=A(K)*X(J)+B(K)*Y(J)+C(K)*Z(J)
670 IF J>1 THEN 700
680 MOVE 1,L(J)
690 MOVE @1:1,L(J)
700 DRAW J,L(J)
710 DRAW @1:J,L(J)
720 NEXT J
730 GO TO K OF 740,760,780,800,820,840,860,880,900,920,940,960
740 A$="I"
750 GO TO 970
760 A$="II"
770 GO TO 970
780 A$="III"
790 GO TO 970
800 A$="AVR"
810 GO TO 970
820 A$="AVL"
830 GO TO 970
840 A$="AVF"
850 GO TO 970
860 A$="V1"
870 GO TO 970
880 A$="V2"
890 GO TO 970
900 A$="V3"
910 GO TO 970
920 A$="V4"
930 GO TO 970
940 A$="V5"
950 GO TO 970
960 A$="V6"
970 PRINT A$
980 PRINT @1:A$
990 NEXT K
1000 END
```

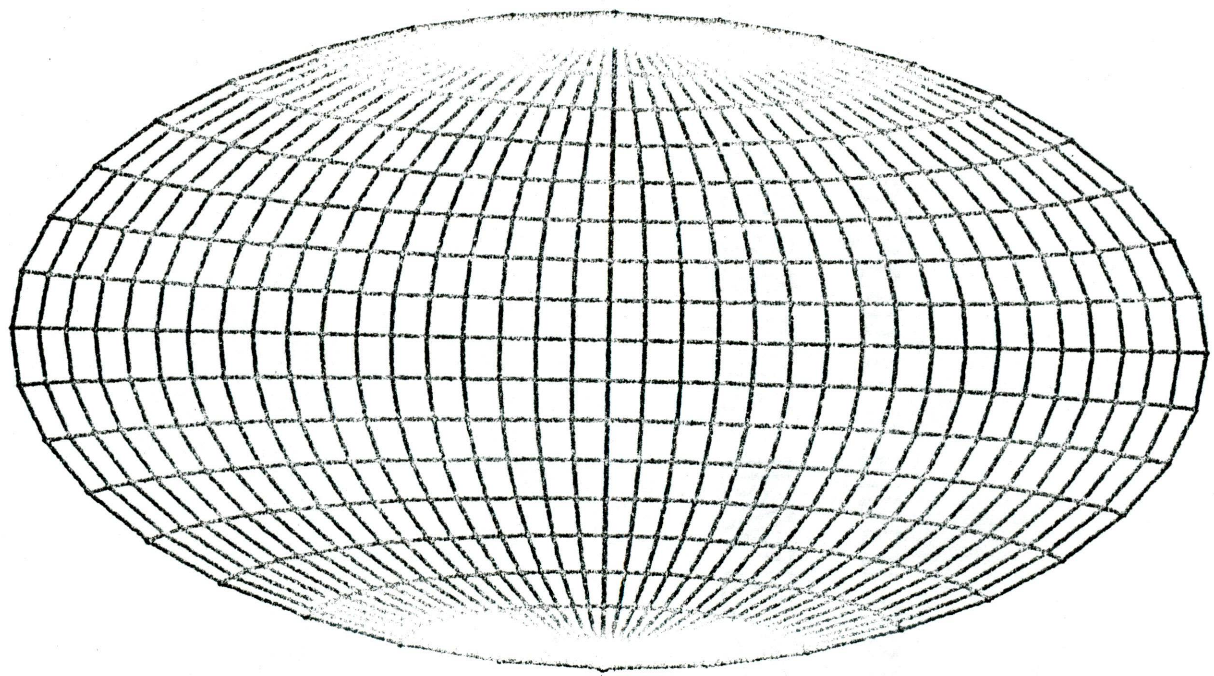


Ex. 6 12 Lead Synthesis (Dower)
Produced by the Tektronix (Basic)

APPENDIX IV

```
100 VIEWPORT 15,115,0,100
110 WINDOW -1.7,1.7,-1.7,1.7
120 FOR K=1 TO 2
130 IF K=1 THEN 200
140 FOR Z=-80 TO 80 STEP 10
150 L=0
160 P=Z*3.14159/180
170 FOR Q=-180 TO 180 STEP 10
180 T=Q*3.14159/180
190 GO TO 250
200 FOR Z=0 TO 180 STEP 10
210 L=0
220 T=Z*3.14159/180
230 FOR Q=-90 TO 270 STEP 10
240 P=Q*3.14159/180
250 X=COS(P)*SIN(0.5*T)
260 Y=SIN(P)
270 S=X^2+Y^2
280 R=(2*(1-(1-S)^(1/2)))^(1/2)
290 IF S=0 THEN 320
300 S=R/S^(1/2)
310 GO TO 330
320 S=0
330 X=X*S
340 Y=Y*S*0.5
350 IF L<>0 THEN 380
360 MOVE X,Y
370 MOVE @1:X,Y
380 DRAW X,Y
390 DRAW @1:X,Y
400 L=L+1
410 NEXT Q
420 NEXT Z
430 NEXT K
440 END
```

EX.1 TEKTRONIX BASIC PROGRAM PRODUCES AITOFF PROJECTION



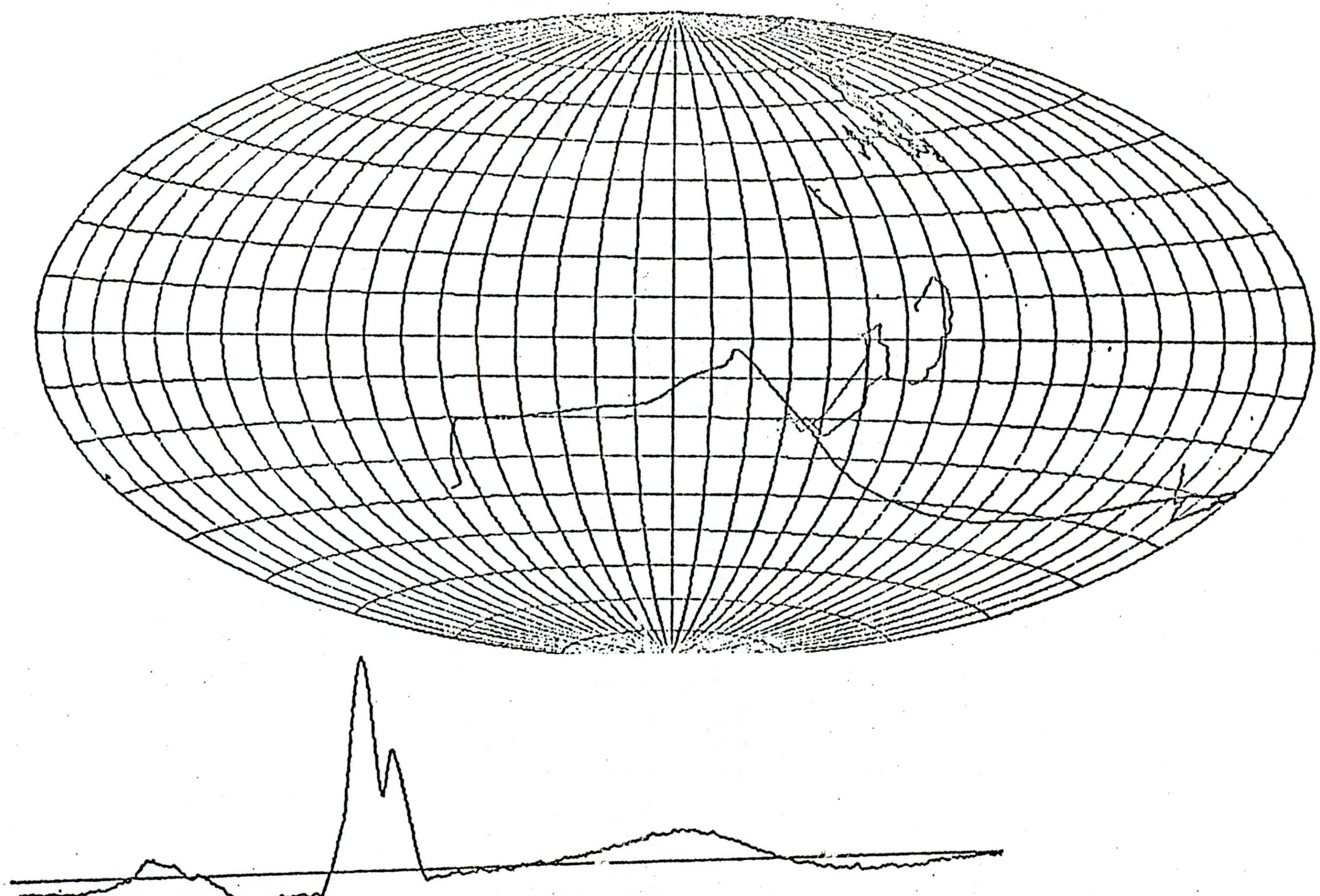
Ex. 2 Aitoff Projection.

Produced by the Tektronix (Basic)

```
      INTEGER T(9984)
      INTEGER BND,EDD,CLOMP
      INTEGER MXYZ(750,3)
      INTEGER A,B,C
      DIMENSION KOL(20)
      DIMENSION R(750)
      DIMENSION KORN(20)
      DIMENSION FIX(3)
      CALL PEFILE (1,1,126*30,IPT,@MEDIAN@)
      CALL FILE (1,1)
      READ (4) ILIM,BND,EDD,CLOMP,MEDRR,(T(I),I=1,ILIM)
      DO 102 I=1,3
102   READ (4) ILIM,FIX,(MXYZ(J,I),J=1,ILIM)
      RMAX=-123
      DO 609 L=1,ILIM
      R(L)=SQRT((FIX(1)*MXYZ(L,1))**2+
     *          (FIX(2)*MXYZ(L,2))**2+
     *          (FIX(3)*MXYZ(L,3))**2)
609   IF (R(L) .GT. RMAX) RMAX=R(L)
      CALL ERASE
      CALL CDRIVE
      CALL WORLD (T,9984)
      PAUSE
      CALL SAVBUF (KOL)
10    WRITE (17,20)
20    FORMAT (@ ENTER LIMITS@)
      READ (15,30) A,B,C,D
30    FORMAT (3I5,F5.0)
      RNAX=D*RMAX
      DO 50 I=1,C
      REVER=(I-1)*RMAX/C
      CALL PUP
      DO 50 J=A,B
      IF (R(J) .GT. RNAX) GO TO 40
35    CALL PUP
      GO TO 50
40    CONTINUE
      IF (R(J) .LT. REVER) GO TO 35
      X=MXYZ(J,1)*FIX(1)
      Y=MXYZ(J,2)*FIX(2)
      Z=MXYZ(J,3)*FIX(3)
      CALL AITOFF (1,ATAN2(Y,X),-ATAN2(Z,SQRT(X*X+Y*Y)))
50    CONTINUE
      CALL PLOT1 (2,0,0)
      CALL DRAW1 (0.,1000.,10.,10.,-.75,.75,3.,10.)
```

EX. 3 EMR 6130 FORTRAN PROGRAM
PRODUCES AITOFF PROJECTION

```
      CALL PUP
      DO 801 J=A,B
801   CALL DRAW8 (1,FLOAT(J),R(J))
      CALL DRAW8 (201,1.,RNAX)
      CALL DRAW8 (102,FLOAT(ILIM),RNAX)
      CALL HRDCPY (T,9984)
      CALL ERASE
      CALL CDRIVE
      CALL PLOT1 (2,0,960)
      CALL FLOOSH
      CALL HRDCPY (T,9984)
      PAUSE
      END
      EOF
```

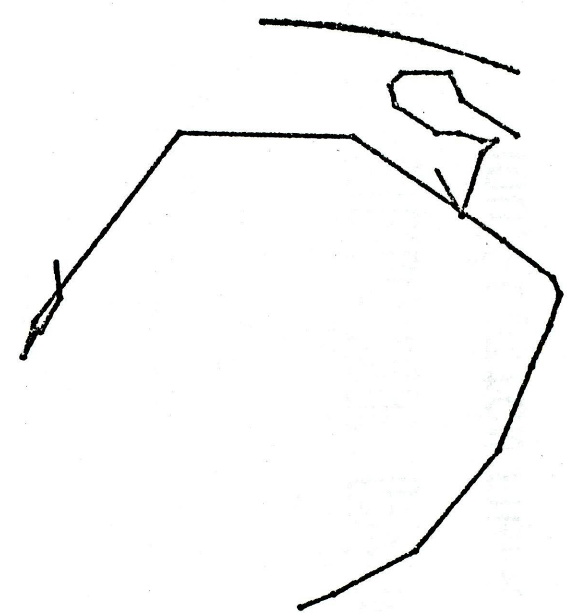


EX. 4 POLAR PLOT ON AITOFF PROJECTION
PRODUCED BY THE 6130 (FORTRAN)

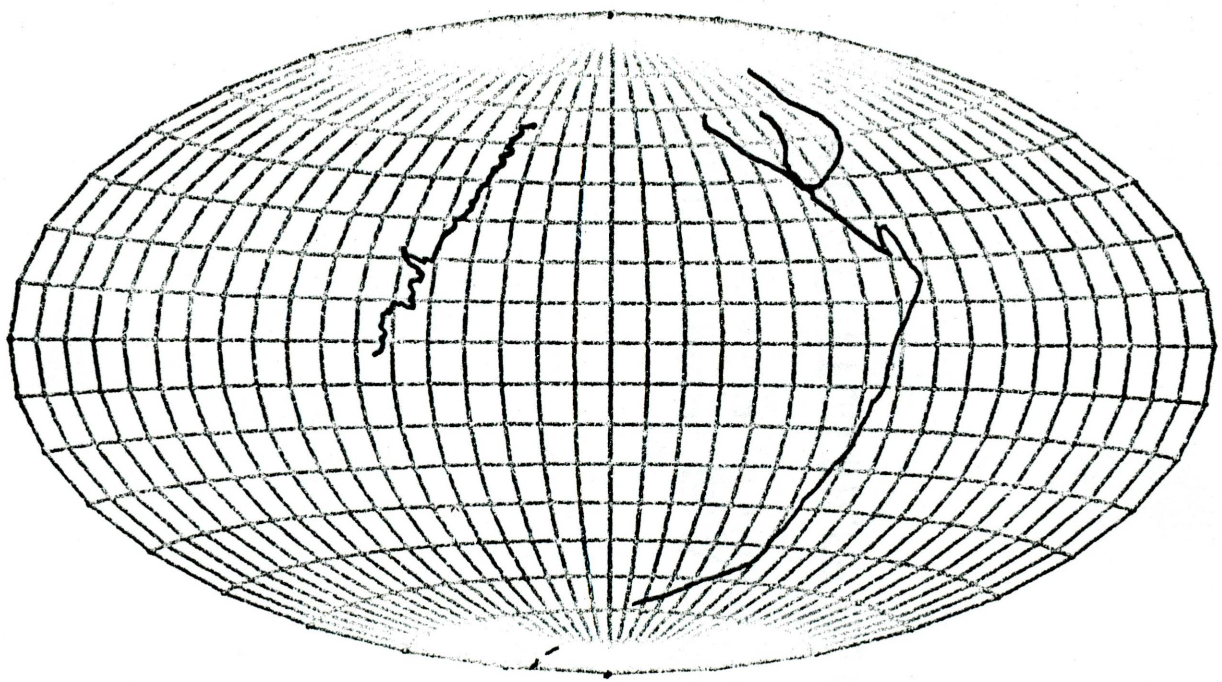
```
100 FIND 2
110 DIM X(200),Y(200),Z(200),T(200),P(200),M(200)
120 DIM B$(10)
130 I=1
140 INPUT @33:X(I),Y(I),Z(I)
150 IF X(I)=0 THEN 210
160 X(I)=X(I)-5000
170 Y(I)=Y(I)-5000
180 Z(I)=Z(I)-5000
190 I=I+1
200 GO TO 140
210 PRINT "ENTER BOUNDRY POINT"
220 INPUT B5
230 PAGE
240 L=0
250 I=I-1
260 WINDOW 0,300,-20,400
270 VIEWPORT 15,115,0,100
280 MOVE 0,B5
290 DRAW 300,B5
300 MOVE 0,0
310 FOR J=1 TO I
320 M(J)=(X(J)^2+Y(J)^2+Z(J)^2)^(1/2)
330 DRAW J,M(J)
340 NEXT J
350 MOVE 0,390
360 PRINT "DO YOU WISH TO USE THIS BOUNDRY LINE"
370 INPUT B$
380 IF B$="NO" THEN 210
390 PAGE
400 FOR J=1 TO I
410 M(J)=(X(J)^2+Y(J)^2+Z(J)^2)^(1/2)
420 IF X(J)<>0 THEN 510
430 IF Y(J)<>0 THEN 460
440 P(J)=0
450 GO TO 540
460 IF Y(J)>0 THEN 490
470 P(J)=-3.14159/2
480 GO TO 540
490 P(J)=3.14159/2
500 GO TO 540
510 P(J)=ATN(Y(J)/X(J))
520 IF X(J)=>0 THEN 540
530 P(J)=-P(J)
540 IF Z(J)<>0 THEN 630
550 IF X(J)<>0 THEN 580
```

EX. 5 TEKTRONIX BAAIC PROGRAM PRODUCES POLAR PLOT

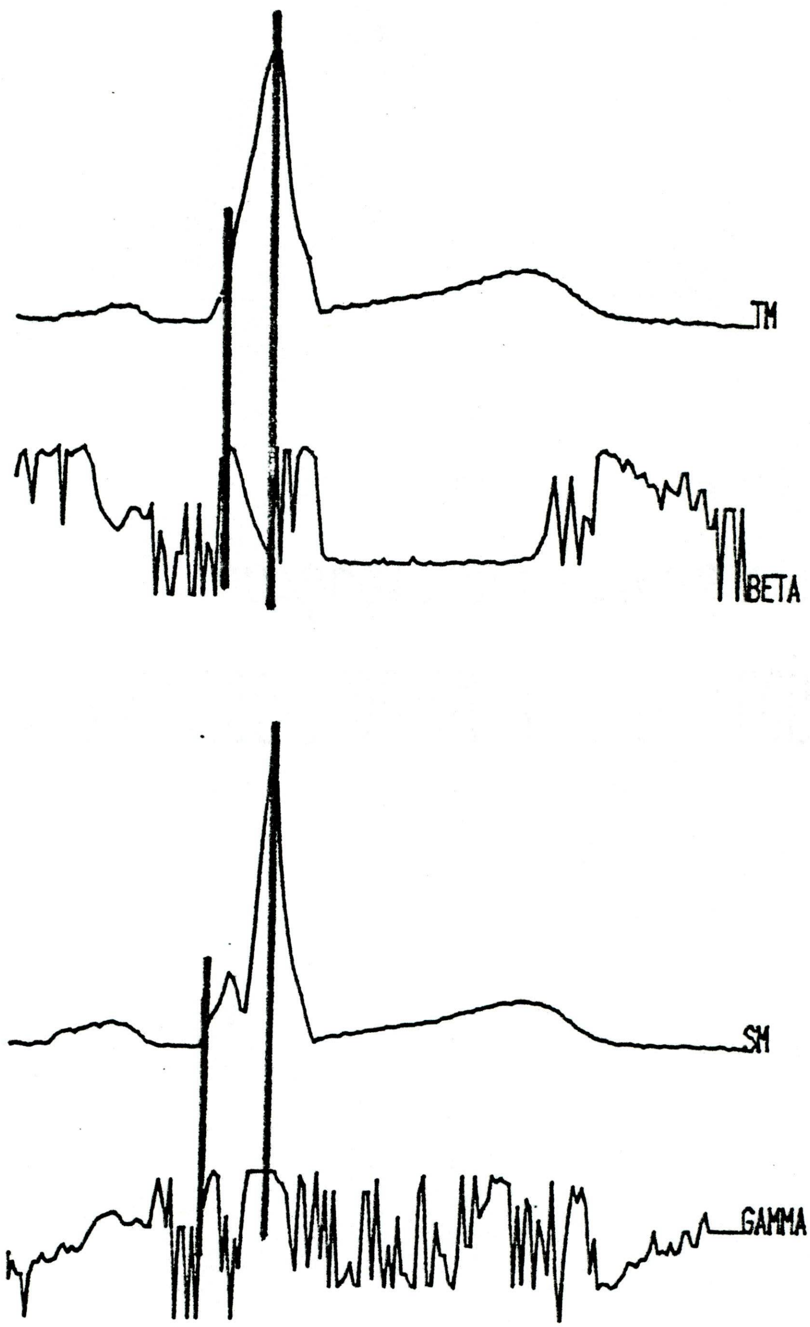
```
560 T(J)=0
570 GO TO 730
580 IF X(J)>0 THEN 610
590 T(J)=-3.14159/2
600 GO TO 730
610 T(J)=3.14159/2
620 GO TO 730
630 IF X(J)<>0 THEN 670
640 IF Z(J)>0 THEN 670
650 T(J)=3.14159
660 GO TO 730
670 T(J)=ATN(X(J)/Z(J))
680 IF Z(J)=>0 THEN 730
690 IF T(J)<0 THEN 720
700 T(J)=T(J)-3.14159
710 GO TO 730
720 T(J)=T(J)+3.14159
730 VIEWPORT 15,115,0,100
740 WINDOW -1.7,1.7,-1.7,1.7
750 P1=P(J)
760 T1=T(J)
770 X1=COS(P1)*SIN(T1/2)
780 Y1=SIN(P1)
790 S=X1^2+Y1^2
800 R=(2*(1-(1-S)^(1/2)))^(1/2)
810 IF S=0 THEN 840
820 S=R/S^(1/2)
830 GO TO 850
840 S=0
850 X2=X1*S
860 Y2=Y1*S*0.5
870 IF M(J)>B5 THEN 920
880 IF L<>0 THEN 910
890 MOVE X2,Y2
900 MOVE @1:X2,Y2
910 IF M(J)<B5 THEN 960
920 DRAW X2,Y2
930 DRAW @1:X2,Y2
940 L=1
950 GO TO 970
960 L=0
970 NEXT J
980 END
```



Ex. 6 Polar Plot
Produced by the Tektronix (Basic)



Ex. 7 Polar Plot on Aitoff Projection
Produced by the Tektronix (Basic)



EX. 8 MAGNITUDE VS. ANGLE WAVES

```
100 DIM A$(10)
110 FIND 2
120 DIM B(200),T(200),G(200),S(200)
130 DIM X(200),Y(200),Z(200)
140 VIEWPORT 15,115,75,100
150 I=0
160 I=I+1
170 INPUT @33:X(I),Y(I),Z(I)
180 IF X(I)=0 THEN 230
190 X(I)=X(I)-5000
200 Y(I)=Y(I)-5000
210 Z(I)=Z(I)-5000
220 GO TO 160
230 PAGE
240 WINDOW 0,I,0,400
250 T1=(X(1)^2+Z(1)^2)^0.5
260 B1=SIN(Z(1)/X(1))
270 S1=(Y(1)^2+Z(1)^2)^0.5
280 G1=SIN(-Y(1)/Z(1))
290 MOVE 1,T1
300 MOVE @1:1,T1
310 FOR J=1 TO I-1
320 T(J)=(X(J)^2+Z(J)^2)^0.5
330 IF X(J)=0 THEN 360
340 B(J)=SIN(Z(J)/X(J))
350 GO TO 370
360 B(J)=3.14159/2
370 S(J)=(Y(J)^2+Z(J)^2)^0.5
380 IF Y(J)=0 THEN 410
390 G(J)=SIN(Z(J)/-Y(J))
400 GO TO 420
410 G(J)=3.14159/2
420 DRAW J,T(J)
430 DRAW @1:J,T(J)
440 NEXT J
450 PRINT "TM"
460 PRINT @1:"TM"
470 VIEWPORT 15,115,50,75
480 WINDOW 0,I,-3.14159,3.14159
490 MOVE 1,-B1
500 MOVE @1:1,-B1
510 FOR J=1 TO I-1
520 DRAW J,-B(J)
530 DRAW @1:J,-B(J)
540 NEXT J
550 PRINT "BETA"
560 PRINT @1:"BETA"
570 VIEWPORT 15,115,25,50
580 WINDOW 0,I,0,400
```

EX. 9 TEKTRONIX BASIC PROGRAM PRODUCES MAGNITUDE VS. ANGLE WAVES

```
590 MOVE 1,S1
600 MOVE @1:1,S1
610 FOR J=1 TO I-1
620 DRAW J,S(J)
630 DRAW @1:J,S(J)
640 NEXT J
650 PRINT "SM"
660 PRINT @1:"SM"
670 VIEWPORT 15,115,0,25
680 WINDOW 0,I,-3.14159,3.14159
690 MOVE 1,-G1
700 MOVE @1:1,-G1
710 FOR J=1 TO I-1
720 DRAW J,-G(J)
730 DRAW @1:J,-G(J)
740 NEXT J
750 PRINT "GAMMA"
760 PRINT @1:"GAMMA"
770 END
```