NOAA TM ERL WMPO-15

UNITED STATES
DEPARTMENT OF
COMMERCE
PUBLICATION

NOAA Technical Memorandum ERL WMPO-15

U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Environmental Research Laboratories

Computer Software for Rainfall Analyses and Echo Tracking of Digitized Radar Data

STELLAN SVEN ÖSTLUND

Weather
Modification
Program Office
BOULDER,
COLORADO
March 1974

ENVIRONMENTAL RESEARCH LABORATORIES

WEATHER MODIFICATION PROGRAM OFFICE

IMPORTANT NOTICE

Technical Memoranda are used to insure prompt dissemination of special studies which, though of interest to the scientific community, may not be ready for formal publication. Since these papers may later be published in a modified form to include more recent information or research results, abstracting, citing, or reproducing this paper in the open literature is not encouraged. Contact the author for additional information on the subject matter discussed in this Memorandum.

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
Environmental Research Laboratories

NOAA Technical Memorandum ERL WMPO-15

COMPUTER SOFTWARE FOR RAINFALL ANALYSES AND ECHO TRACKING OF DIGITIZED RADAR DATA

Stellan Sven Östlund

Experimental Meteorology Laboratory

Weather Modification Program Office
Boulder, Colorado
March 1974

CONTENTS

COMPUTER SOFTWARE FOR RAINFALL ANALYSES

AND ECHO TRACKING OF

DIGITIZED RADAR DATA

by

Stellan Sven Östlund

ABSTRACT

This paper describes computer software designed for digitized radar data. Program packages include a scan-conversion from a polar to a Cartesian grid system, a rain summation analysis over selected areas within the whole area, and an echo tracking program which calculates total rainrates and rainfalls from isolated echoes matched from frame to frame. All the results may be drawn on a pen plotter for easier interpretation.

To avoid confusion, four definitions should be made as follows:

1) Rainrate refers to rainfall rate at a point, measured in mm/hr.

2) Volume rainrate refers to rainrates integrated over an area to give rain volume per time (m^3/hr or mm-sq mile/hr).

3) Total rainfall is rainrates integrated over time to give depth of rainfall (mm).

4) Total volume rainfall is rainrates integrated over both time and area to give volume rain (m^3 or mm-sq mile).

RSUM prints total rainfall in millimeters over selected target areas during selected time periods for one day of data. Total volume rainfall for the whole target area is also calculated.

TRACK isolates echoes at a specified rainrate threshold and finds the volume rainrate for these echoes. The program then tracks these echoes through time and integrates the volume rainrates of each echo to give total volume rainfall for each echo. The frame by frame analysis may be printed either on a pen plotter or CRT display.

I. INTRODUCTION

Data to be analyzed is typically of range normalized rainrates supplied at 2^o azimuth increments and one-half mile range increments previously unpacked from a radar digitizer. A report by Wiggert and Andrews (1974) will be forthcoming on the digitizer system. Since data in the northwest quadrant was primarily of interest, azimuth readings were only in the range 270^o to 360^o. Thus, 45 azimuths, each with 300 bins were to be analyzed each sweep, one sweep recorded every five minutes.

To save computer core during the scan conversion, a method was used which stores only two adjacent azimuths at a time, and which reads these points into a portion of a 73 x 62 Cartesian grid array representing a 72 n mi x 61 n mi target area to be analyzed (fig. 1). The data can then be more easily processed and printed out from Cartesian coordinates. In subsequent programs data summary packages include echo isolation, Fourier analysis of echo contours, echo tracing and calculating echo volume rainrates, as well as echo tracking through time.

To convert data from B-Scan to Cartesian coordinates requires about 15 seconds of computer time for each 90^o scan on a UNIVAC 1106 computer. If repeated analysis of data is anticipated, we have found it advantageous to convert the whole day's data into Cartesian coordinates and write this out on tape before proceeding with the other analyses. This avoids reconverting data to Cartesian each time. This Cartesian data can then be entered into either the tracking program (TRACK), or the rainfall integration program (RSUM).

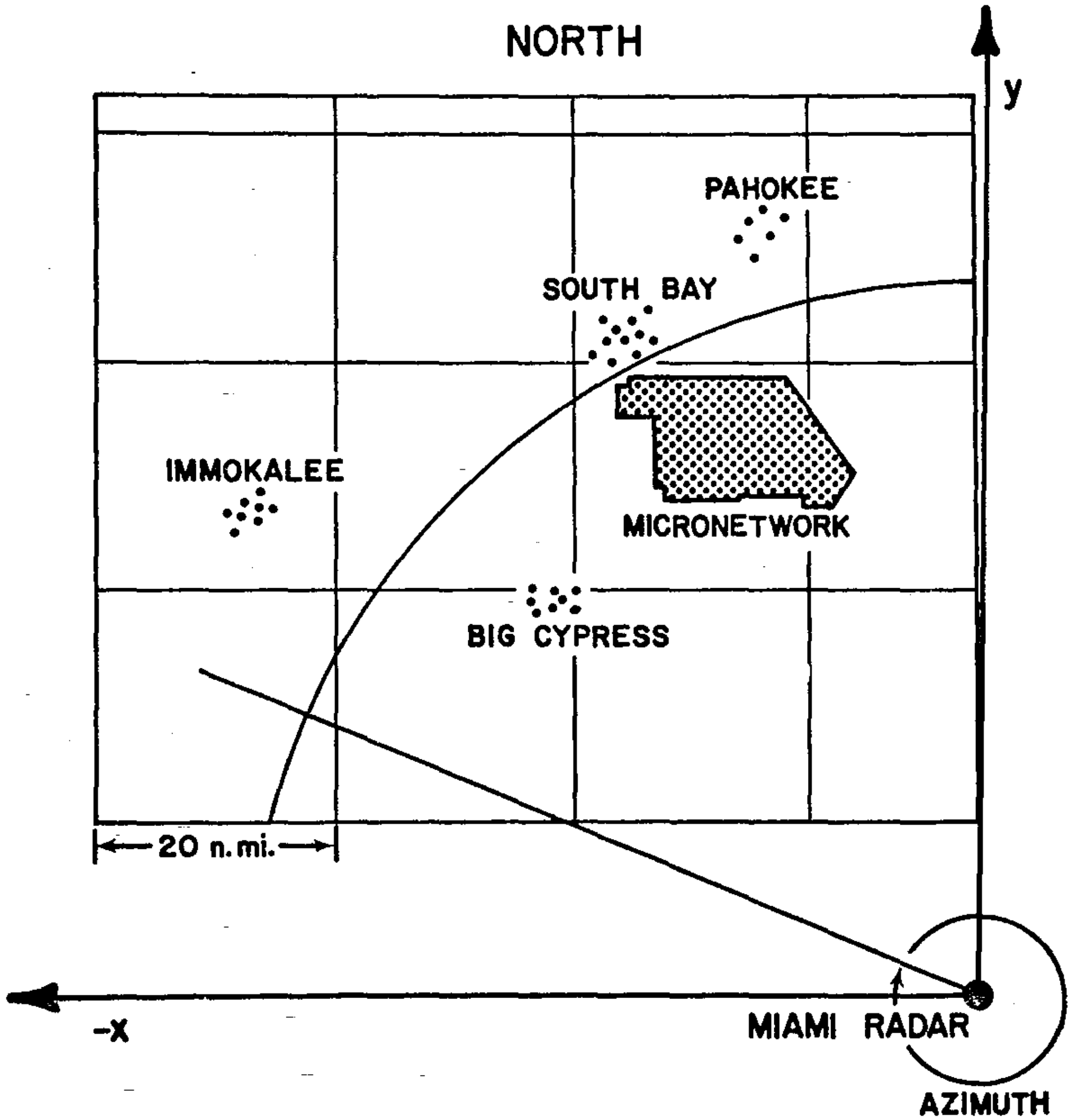


Figure 1. Geometry of Target Area.

2. B-SCAN TO CARTESIAN DATA CONVERSION (KART)

2.1 Background

KART converts data from polar coordinates into Cartesian coordinates. Data is supplied from tape and is usually at one-half mile 2° intervals, although these parameters are variable in the program. The radius of the data from the radar is between 15 n mi and 80 n mi, with the result that the data density ranges from 3.8/n mi^2 to .72/n mi^2. Therefore, a 1 x 1 mile square grid system seemed appropriate, although at the periphery the grid will overestimate the accuracy of the available data, while information will be lost at small radius. The computed Cartesian data is written on tape.

Previous practice has been to read all the polar data into the nearest grid point on the rectangular grid. The rectangular array is then searched for "holes" where no data has been supplied at larger radius. These holes are then filled using the mode, max, min, or average of the surrounding points. (See, for instance Booth, 1972). This method is fast, but has the disadvantage of misrepresenting data over small area at large radii. In addition, an uncertainity of $\pm$ 1 mile in locating the data point will result. Thus, a more accurate interpolation scheme was designed. This method is described in the next section.

2.2 General Method of Scan Conversion

The data from two adjacent angles θ_1 and θ_2 are read simultaneously. All grid points in the sector $\theta_1 - \theta_2$ are found. Let (X_o, Y_o) be a grid location from the radar origin. Let (X_z, Y_z) be the displacement of the grid origin from the radar origin, and let (X_i, Y_i) be the grid location from the grid origin (fig. 2). The relation

$$\tan\theta_1 \leq Y_o/X_o \leq \tan\theta_2$$

is necessary for (X_o, Y_o) to be in the sector. A linear interpolation for the value at (X_o, Y_o), (location (X_i, Y_i) in the grid) is performed from the four surrounding polar data points.

Let g be the rainrate to be interpolated in the grid system. Pij is the rainrate at the indicated point

$$g(X_i, Y_i) = G + t(F - G)$$

where

$$G = P_{11} + S\,(P_{21} - P_{11})$$

$$F = P_{12} + S\,(P_{22} - P_{12})$$

and

$$S = (\tan^{-1}\left(Y_o/X_o\right) - \theta_1)/(\theta_2 - \theta_1)$$

$$t = (\sqrt{X_o^2 + Y_o^2} - r_1)/(r_2 - r_1)$$

$$X_o = X_z + N_i\delta \qquad Y_o = Y_z + M_i\delta$$

$$X_i = N_i\delta \qquad Y_i = M_i\delta$$

where M_i refers to the column of (X_i, Y_i)

N_i refers to the row of (X_i, Y_i)

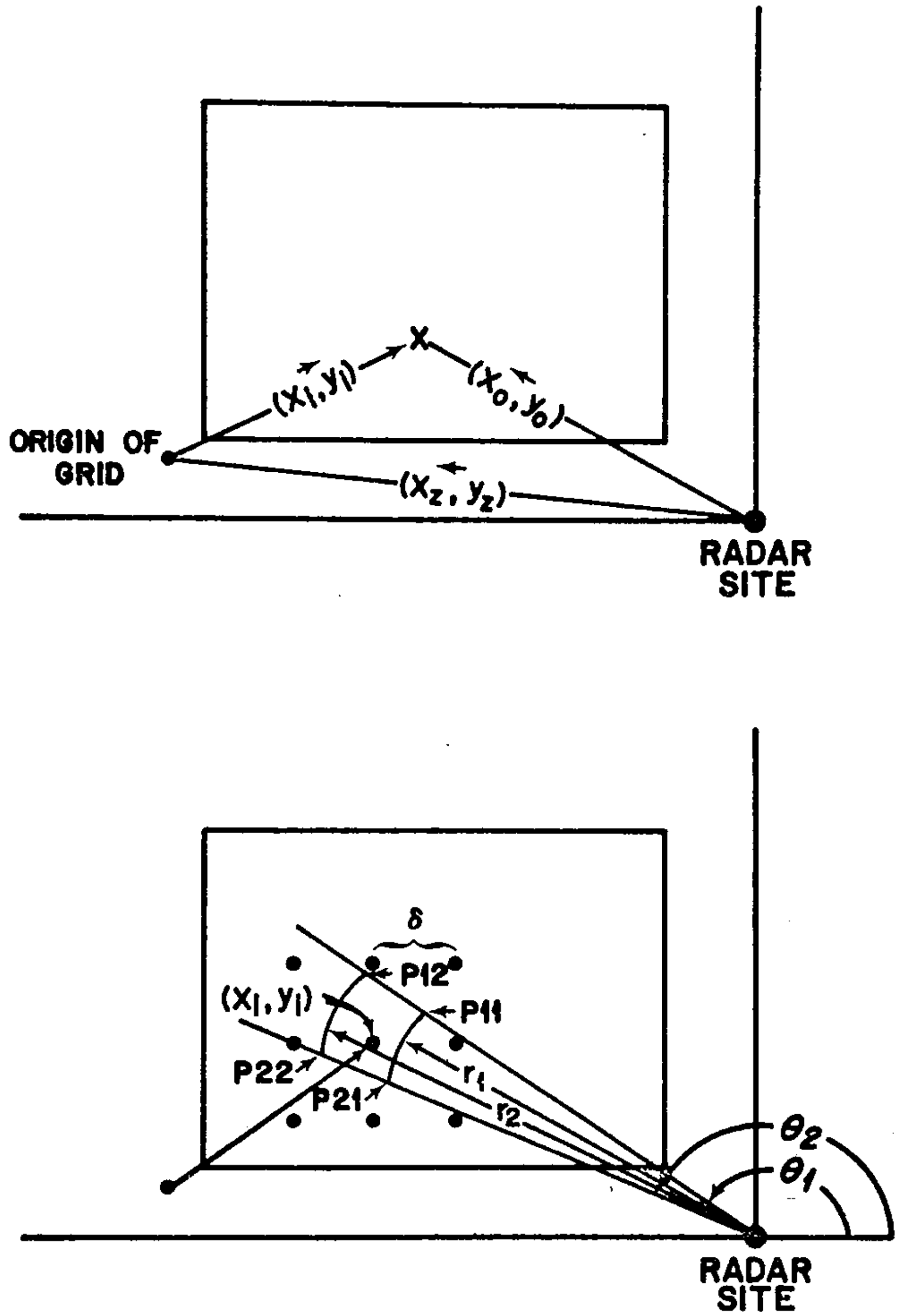


Figure 2. Interpolating Rectangular Grid Point Values from Polar Coordinate Values

$$\tan\theta_1 \leq Y_o/X_o \leq \tan\theta_2$$

$$\cot\theta_2 \leq X_o/Y_o \leq \cot\theta_1$$

$$Y_o \cot\theta_2 \leq X_o \leq Y_o \cot\theta_1$$

$$Y_o \cot\theta_2 \leq X_z + N_i\delta \leq Y_o\cot\theta_1$$

and performing similar substitution for Y_o

$$\frac{(Y_z + M_i\delta)\cot\theta_2 - X_z}{\delta} \leq N_i \leq \frac{(Y_z + M_i\delta)\cot\theta_1 - X_z}{\delta}$$

This is a necessary condition that $g\ (X_i,\ Y_i)$ can be interpolated from the data on the two azimuths and is sufficient in the northwest quadrant.

2.3 Calling Sequence and Internal Options for KART.

Parameters to be specified in a COMMON statement with the main program, only for internal purposes:

IBINS	-	number of bins of data along one azimuth
IN	-	I dimension (X axis) of rectangular array.
JN	—	J dimension of array
DIS	-	Distance between grid points
XZ	-	X-coordinate or <u>origin</u> of rectangular grid (lower left corner of the array is (1,1), not the origin.)
YZ	-	Y-coordinate of origin of grid
KCART	-	0 at beginning of program. KCART will then count how many times KART has been called. KCART should be reset to zero if the continuity of read data is broken.

BN1(300), BN2(300) - arrays containing data along adjacent azimuths.

Input parameters in calling sequence:

SMT - starting time in minutes. Records are scanned until SMT is exceeded, then scan conversion begins.

EMT - ending time.

Output parameters in calling sequence:

RESP(IN, JN) - output array

KEY = 0 if SMT < TIME < EMT

= 1 if EMT < TIME

= 2 if elevation angle bad.

Output parameters in COMMON:

TDIFF - difference in hours between this frame and preceding frame.

TDA1 - Julian day.

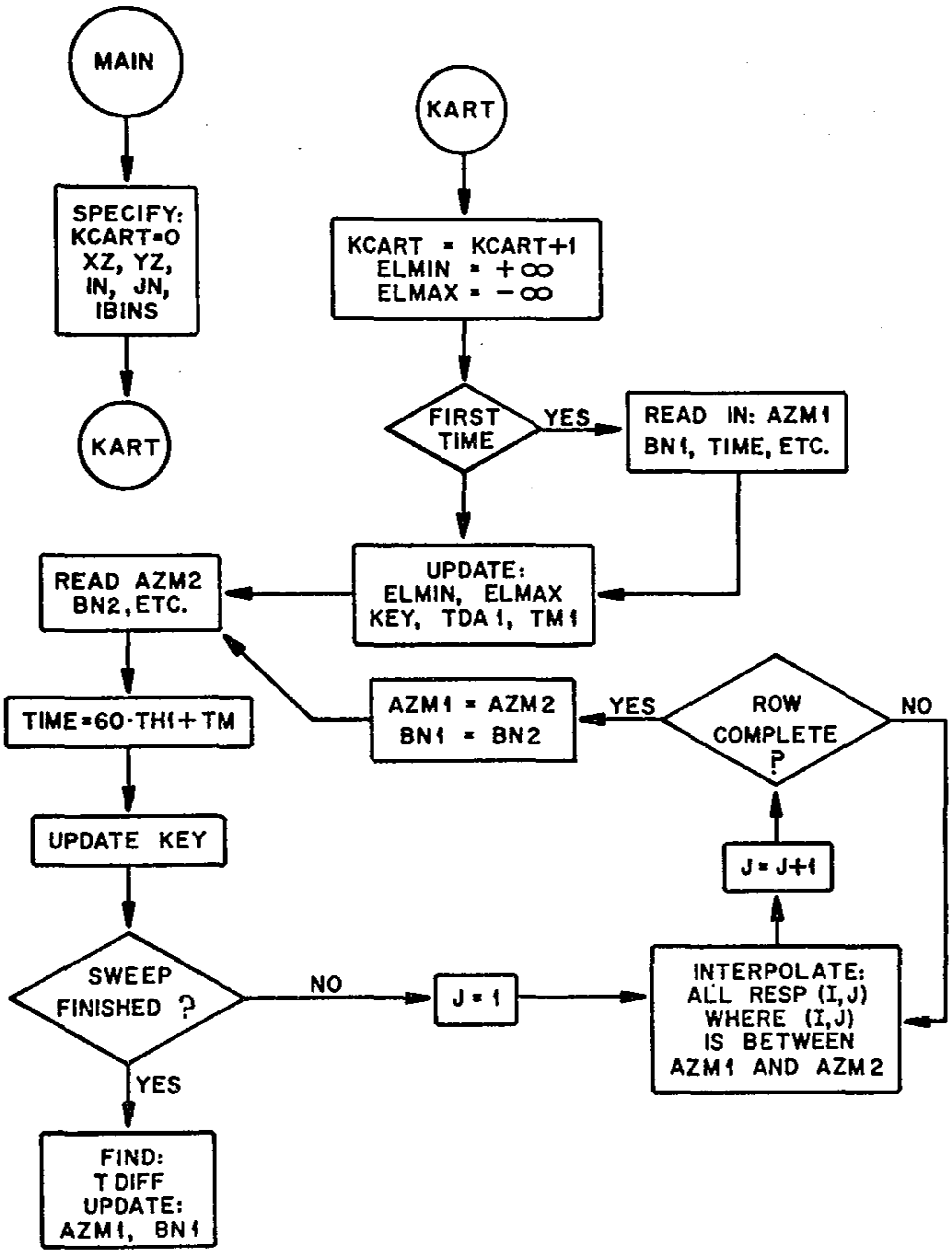
TH1 - hour.

TM1 - minute.

ELMIN - minimum elevation angle of antenna.

ELMAX - maximum elevation angle.

2.4 FLOW CHART SUMMARY



2.5 Map Display (MAPP)

MAPP is a simple printing routine. The program prints the full X-Y array, specified in the call sequence.

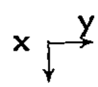
Calling parameters: Z (IE, JE) array to be printed.

All data in the array is scanned for maximum and minimum; then scaled SF = 10/(max-min). Let RSF = 1/SF. Assuming an array whose values are between zero and 20 has been entered, SF = .5, and RSF = 2.0; a printed digit 3 will represent data magnitude from 6.0 to 8.0. A dot (.) represents data between 1.0 and the minimum data threshold for a numeric representation. Example: RSF = 3.1, MIN = 0.

True value = 3.1 x Print value + 0.

Printed Symbol	.	1	2
Represented Value	1. - 3.1	3.1 - 6.2	6.2 - 9.3

Example of printed array on following page (fig. 3); X axis is oriented as follows:



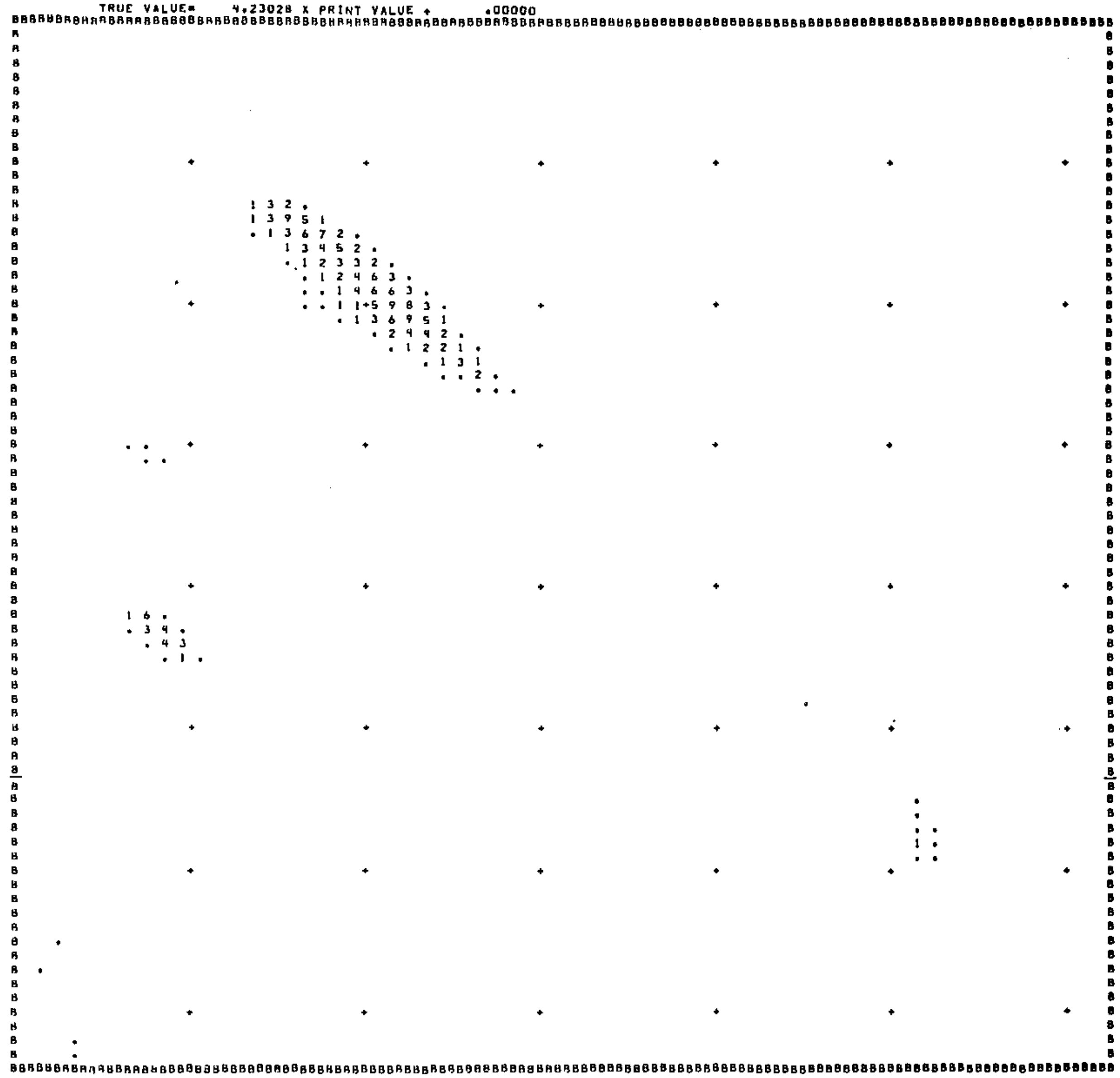
This orientation is rotated 90° from standard to enable the dimensions of the arrays used to fit on one page.

2.6 Sample Printout

EML (FACE, 1973) / KART / OSTLUND DATE 100573 PAGE 137

MINIMUM VALUE= .00000 MAXIMUM VALUE= .42303+02 DIFFRENCE= .42303+02

TRUE VALUE= 4.23028 X PRINT VALUE + .00000



MAX RADAR ELEV .5 MIN RADAR ELE .4

DAY 177.0 HOUR 17.0 MINUTE 5.0

SAMPLE PRINTOUT FROM MAPP ***NORTH*** ⟶

Figure 3.

3. RAIN SUMMATION ANALYSES (RSUM)

3.1 General Method

RSUM is a program which uses tape data in a Cartesian grid generated by KART. Data is analyzed over specific sections of the grid corresponding to the location of the raingage clusters

Six arrays are used in the program: RINT, FIM, SOU, CYP, PAH and GAR. They represent the rainfall volume calculated in a section of the grid which contains the raingage clusters. RINT includes the entire target area, and the others describe the Immokalee, South Bay, Big Cypress, Pahokee and Garstang (University of Virginia) micronetwork areas respectively (figs. 4, 5a, 5b and 6).

On and off time for the summation is specified on six data cards. Up to five non-overlapping periods of time may be analyzed for each cluster. Thus, if the second data card is

```
17.30   18.00   18.30   20.00
```

the Immokalee cluster will be summed between 17:30 and 18:00, results printed, and a new analysis will be repeated between 18:30 and 20:00. The data cards must follow the order specified above. In addition, total rain volume for the entire target area is calculated, and the sum is printed at hourly intervals from the starting time. The geometry of the clusters is illustrated in fig. 1.

A contouring package (CLOT) has been included to give a contoured description of the arrays by use of the Cal-Comp pen plotters. The contour intervals will be adjusted to give approximately six contours. CLOT is neither listed, nor described.

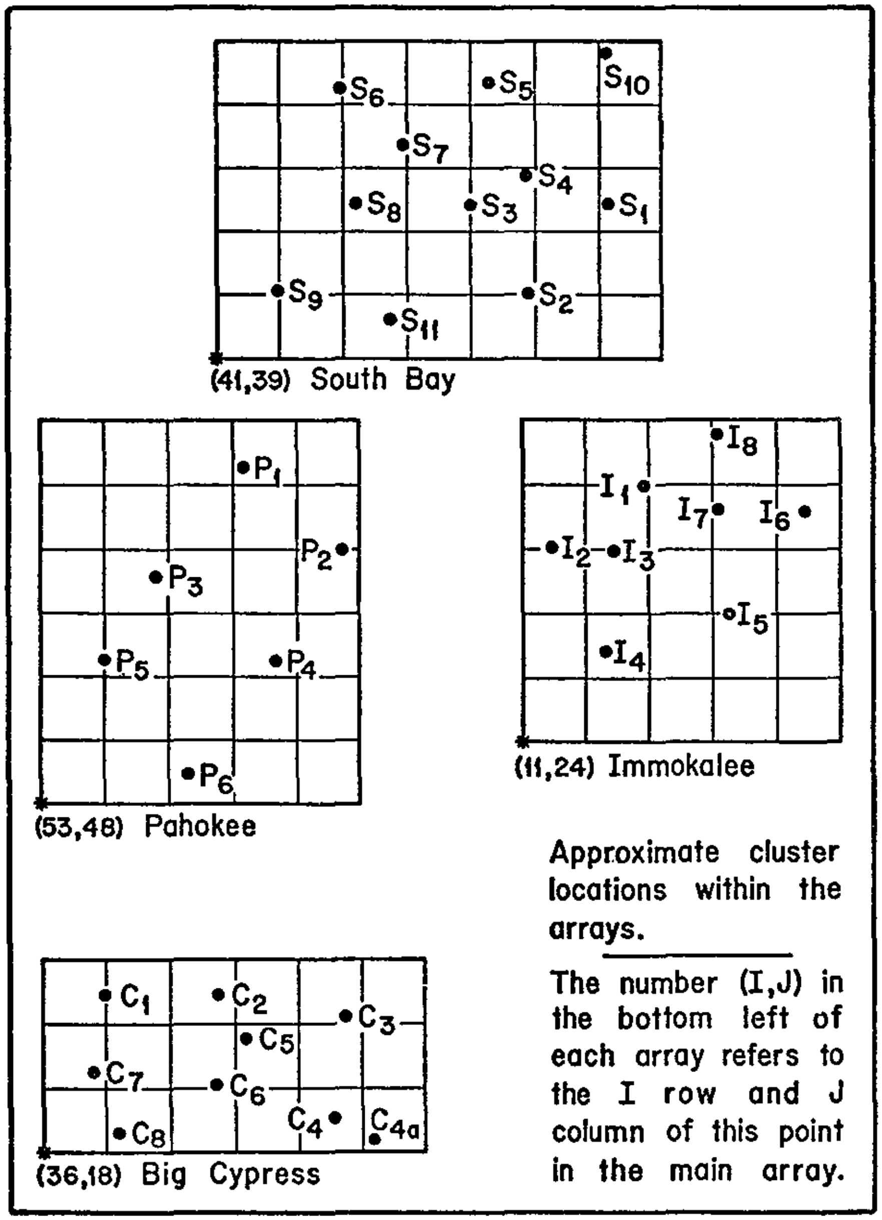


Figure 4. Approximate Cluster Locations within the Arrays

3.2 Sample Printouts

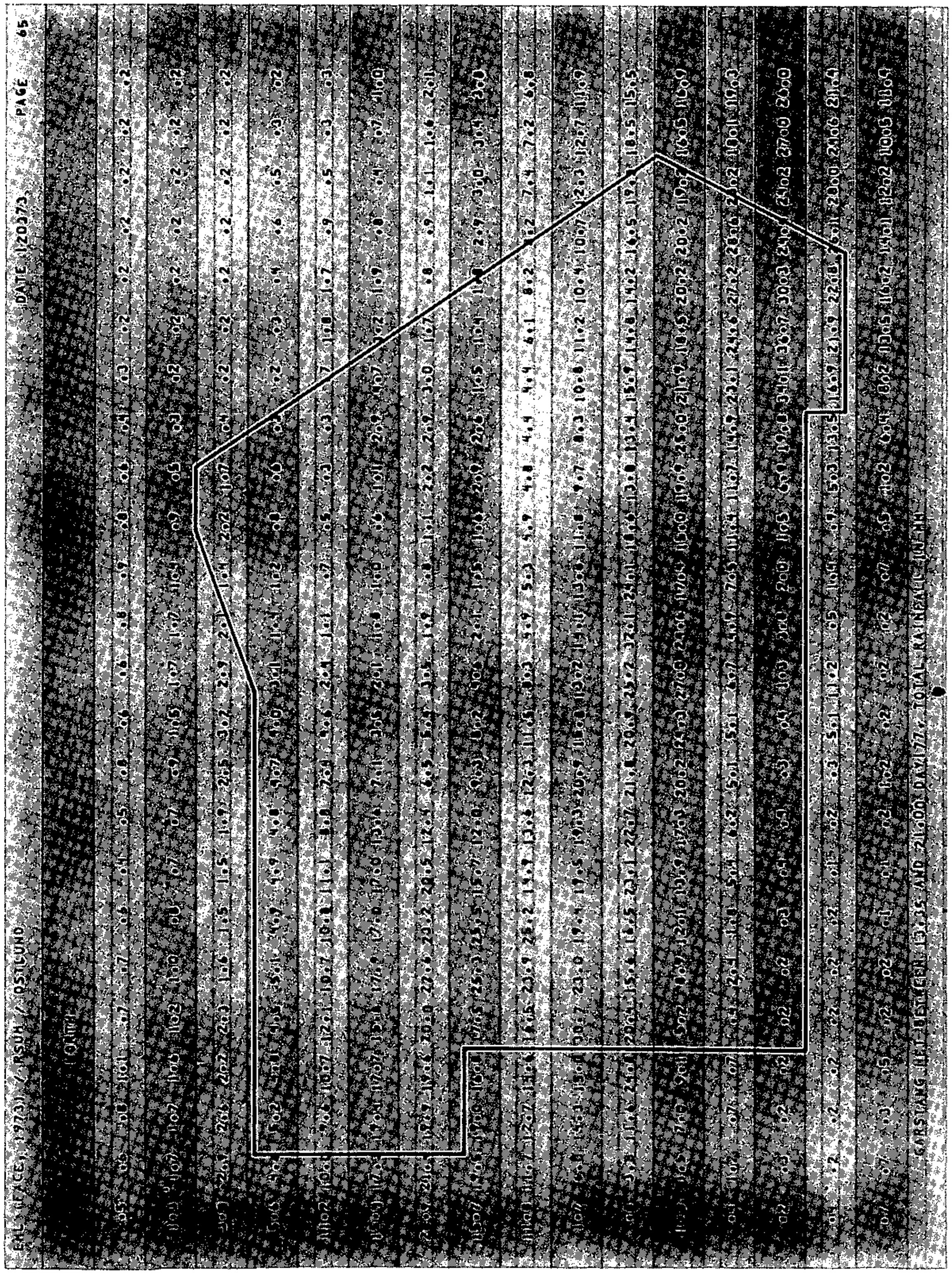


Figure 5a. Garstang (University of Virginia) Micronetwork Rain Depth

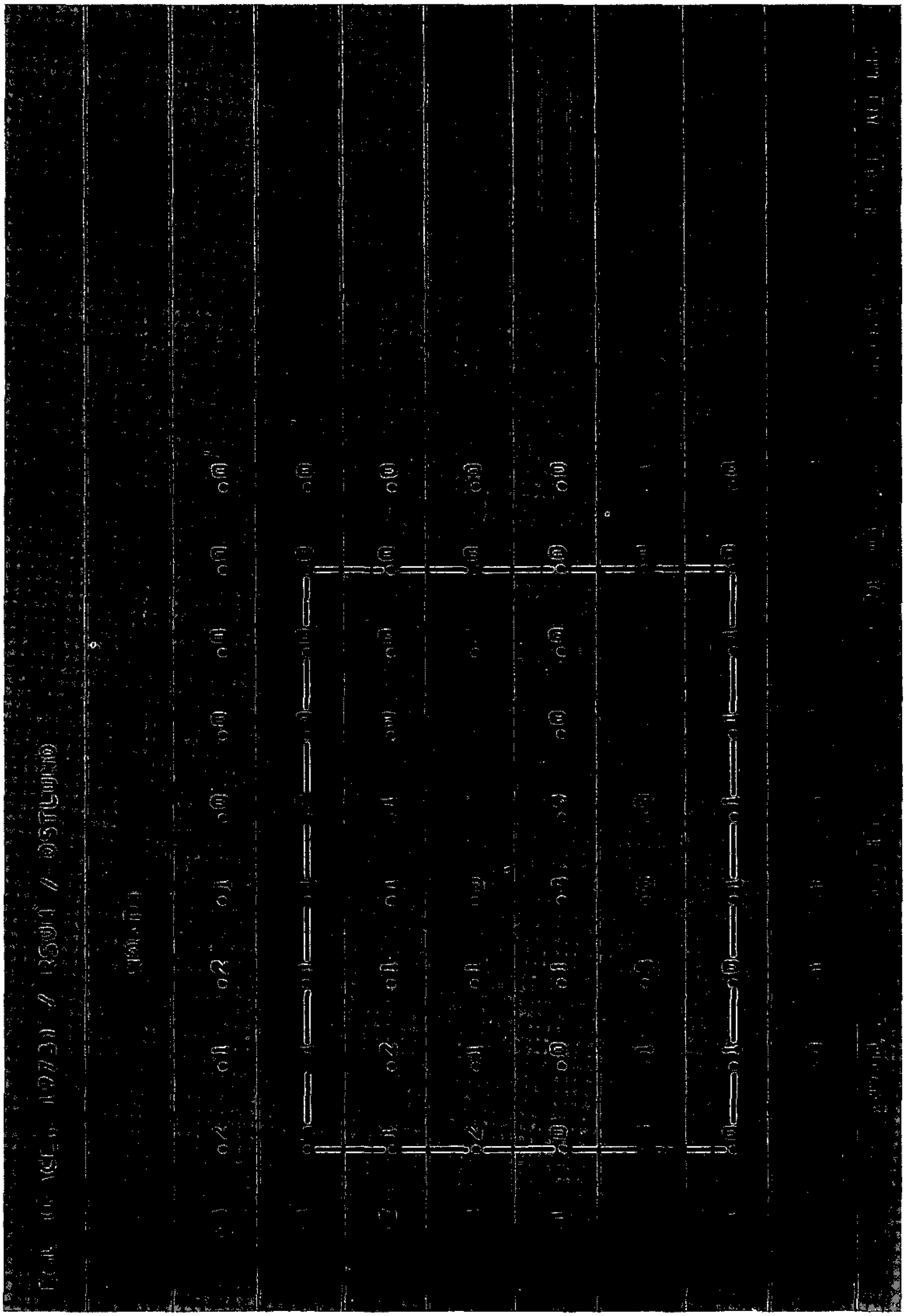


Figure 5b. South Bay Network Rain Depth

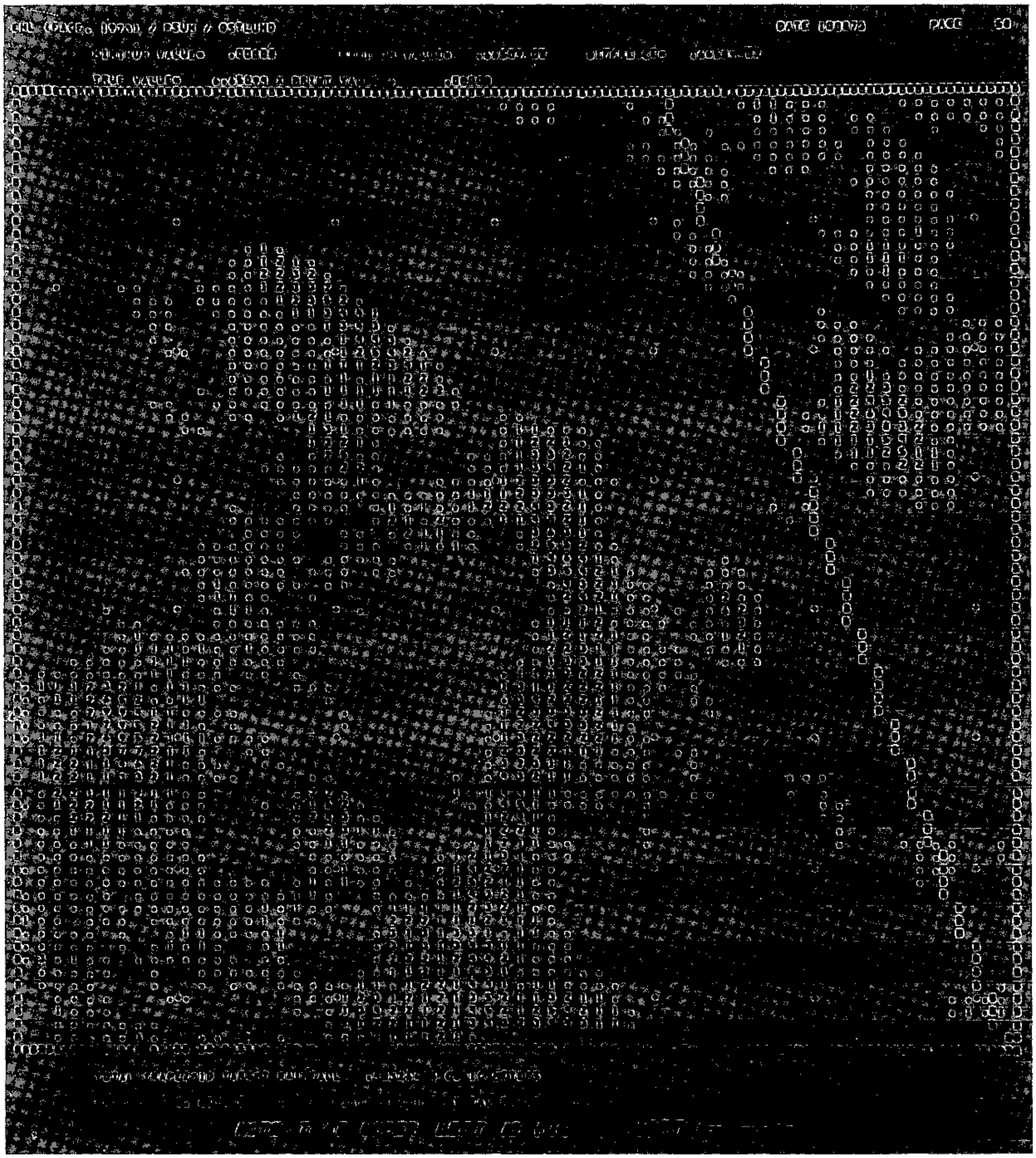


Figure 6. Sample Printout of Rain Summation Analysis

4. ECHO ISOLATION AND ANALYSIS (FOURAN)

4.1 General Method

FOURAN is a program which uses data specified in the form of a rectangular array to isolate individual echoes and to analyze them. This is done in the following steps:

a) The starting point of a contour associated with the threshold is found.

b) The contour is traced through to its endpoint.

c) The echo is defined as the interior of the closed contour. No extrapolation of contours that touch a grid boundary is attempted.

d) Total volume rainrate of the echo (volume/time) is calculated.

e) The path found in b) is harmonically analyzed. If NHAM is the number of harmonics analyzed, then the first NHAM sine and cosine coefficients are stored in an array COF.

f) By formula 3, Section 4.5, applied to the coefficient array in e), the area of the echo, the centroid (of the contour) and the perimeter are evaluated for each echo.

Output is in the form of Fourier coefficients, total area, total volume rainrate, perimeter, and centroid location for each echo. In addition, a Cal-Comp plot of the PPI is obtained.

4.2 Calling Sequence of FOURAN.

Input parameters:

Z (IN, JN) - array to be analyzed. (See note below.)

THRESH - echo threshold.

NHAM - number of harmonics to be used. If no contour plotting is necessary, NHAM = 2 to save unnecessary calculations.

NECH - maximum number of echoes.

Output parameters:

COF (NHAM, 4, NECH) - array of Fourier coefficients.

PERIM(NECH) - array of perimeters of echoes

RFALL(NECH) - array of total rainfalls.

IBRDS (NECH) - starting border of echoes.

AREA(NECH) - area of echoes.

IECH - number of echoes actually found.

A plotting Common should be included if pen plotting is done internally. Note: The borders of the array Z are set to a large negative value to close all contours. Therefore, to avoid losing data, the dimensions of Z should be two higher than the desired array and the actual data points will be in rows 2 through IN-1 and columns 2 through JN-1. No points will be plotted outside this interior array.

Subroutines called in FOURAN:

FOLL - follows path of contour.

PUT - stores points already traced.

DON - (logical function) checks if a contour is completed.

ECHOF - calculates Fourier coefficients.

RF - (function) calculates total rainfall.

Process for defining a contour:

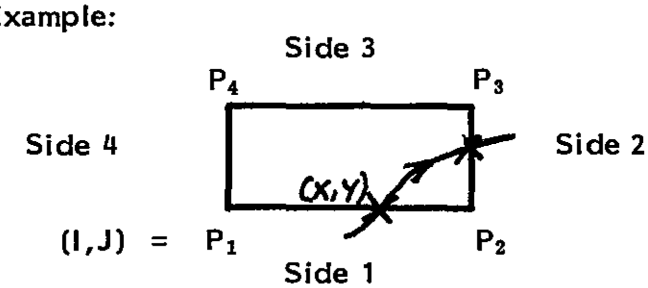
a) The threshold is subtracted from all elements in Z.

b) To find the starting point of a contour, the array is searched, completing each column from right to left, for a contour crossing. A contour crossing is found between (I,J) and (I+1,J) if $Z(I,J) \cdot Z(I+1,J) \leq 0$. If the point has been previously drawn the next starting point is found.

c) A box is defined as a set of four adjacent grid points (P_1, P_2, P_3, P_4). P_1=(I, J), P_2= (I+1,J), P_3=(I+1,J+1), P_4= (I,J+1). The location of a box is defined as at (I, J).

The box corresponding to a grid crossing is found, and the side of the box corresponding to the crossing is labeled.

Example:



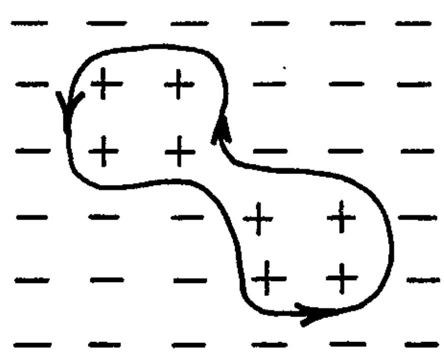
(X,Y) occurs in Box (I,J) on Side 1. The contour continues and the next point occurs on Side 2.

When two points have been found between which a contour crossing occurs, the exact location X and Y is stored in the array XS and YS. The exact location is defined by a linear interpolation between the two adjacent points. The next point of the contour is found by searching the sides of the box in a counterclockwise fashion, starting from the side of entry.

When the next point is found, a new box is redefined appropriately as a check to see if the contour is done. If not, the process is iterated.

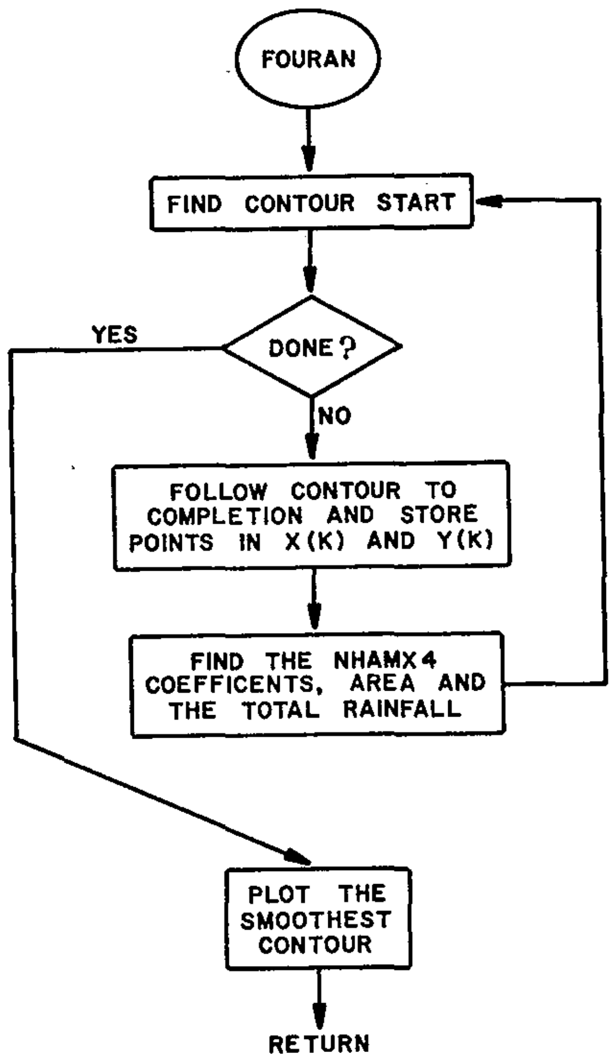
A possible ambiguity occurs at a point where two boxes touch only at a corner. The program includes as much area as possible in each echo, so a corner will not split the echoes.

Example: Contour value is 0.



d) After the contours have been traced through, the XS (I), and YS (I) coordinates are entered into the ECHOF routine to find the Fourier coefficients.

4.3 FLOW CHART OF ECHO ISOLATION AND CONTOUR TRACING



4.4 Volume Rainrate Calculation for Isolated Echoes

Total rainfall is calculated by the RF function in the following way:

A point definitely within the contour is entered and the eight surrounding points are searched to see if they are within the echo. If a surrounding point (I,J) is inside, (I,J) is stored, and Z (I,J) added to the sum. Then the next point's eight surrounding points are checked, similarly, unless the point has been previously counted. The process is iterated until no unprocessed adjacent points within the contour are left. This method is similar to the method used by Gray in an NSSL echo isolation program referenced by Barclay and Wilk, 1970.

4.5 Mathematics of the Fourier Summary

After the XS (I) and YS (I) coordinates are entered into ECHOF, the Fourier coefficients are computed as follows:

Let $\vec{r}(I) = (XS(I), YS(I))$ define the known points along the curve. The true path is defined as the continuous and piecewise linear curve connecting $\vec{r}(I)$ and $\vec{r}(I+1)$. Let s be the arclength parameter from the starting point. Let $(X(s), Y(s))$ be the true path as a function of arclength.

$$(1)\quad X(s) = \sum_{n=o}^{\infty} a_n \cos\left(\frac{2n\pi s}{L}\right) + b_n \sin\left(\frac{2n\pi s}{L}\right)$$

$$(2)\quad Y(s) = \sum_{n=o}^{\infty} c_n \cos\left(\frac{2n\pi s}{L}\right) + d_n \sin\left(\frac{2n\pi s}{L}\right) ,$$

where L is the perimeter of the curve. Note that

$$L = \sum_{I=1}^{IN-1} \Delta s_I \quad \text{where}$$

$$\Delta s_I = \left[(XS(I)-XS(I+1))^2 + (YS(I) - YS(I+1))^2 \right]^{\frac{1}{2}}$$

and for $n \neq o$

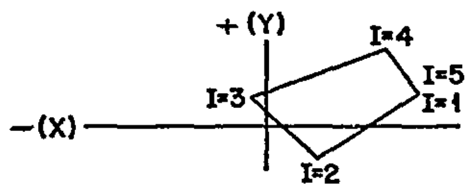
$$a_n = \frac{2}{L} \int X(s) \cos\left(\frac{2n\pi s}{L}\right) ds$$

$$b_n = \frac{2}{L} \int X(s) \sin\left(\frac{2n\pi s}{L}\right) ds$$

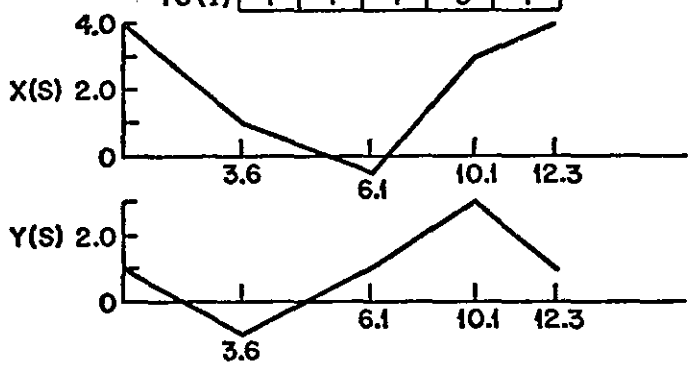
$$c_n = \frac{2}{L} \int Y(s) \cos\left(\frac{2n\pi s}{L}\right) ds$$

$$d_n = \frac{2}{L} \int Y(s) \sin\left(\frac{2n\pi s}{L}\right) ds$$

EXAMPLE:



	1	2	3	4	5=IM
XS(I)	4	1	-.5	3	4
YS(I)	1	-1	1	3	1



By Fourier's theorem (with $1 \leq n \leq$ (NHAM))

$$a_n = \frac{2}{L} \int_0^L X(s) \cos\left(\frac{2n\pi s}{L}\right) ds$$

$$= \frac{2}{L} \sum_{i=1}^{IN} \int_{s_i}^{s_{i+1}} X(s) \cos\left(\frac{2n\pi s}{L}\right) ds$$

$$= \frac{2}{L} \sum_{i=1}^{IN} \int_{s_i}^{s_{i+1}} (p_i + q_i s) \cos\left(\frac{2n\pi s}{L}\right) ds$$

$$= \frac{2}{L} \sum_{i=1}^{IN} \left[\frac{p_i L}{2n\pi} \sin\left(\frac{2n\pi s_i}{L}\right) + \frac{q_i L^2}{(2n\pi)^2} \cos\left(\frac{2n\pi s_i}{L}\right) + \frac{s_i L}{2n\pi} \sin\left(\frac{2n\pi s_i}{L}\right)\right]_{s_i}^{s_{i+1}}$$

where p_i and q_i are coefficients which describe the line segments of the path.

These latter expressions (1) and (2) are easily generated on the computer.

For $n = o$

$$a_o = \frac{1}{L} \int_0^L x(s)\, ds$$

$$= \bar{x} \text{ centroid of curve.}$$

Similarly, b_n, c_n and d_n are found.

Note that a_o and c_o give centroid of the <u>curve</u> for the i^{th} echo.

$$COF(n, 1, I) = a_{n-1}$$

$$COF(n, 2, I) = b_{n-1}$$

$$COF(n, 3, I) = c_{n-1}$$

$$COF(n, 4, I) = d_{n-1}$$

Area of the echo is found by the following method:

$$\text{Area} = \int_A da = \int_A dx\, dy$$

$$= \int_{\partial A} x\, dy$$

$$= \int_{\partial A} x \frac{dy}{ds}\, ds$$

Using (1) and (2),

$$\int_A da = \int_0^L x(s)\frac{dx(s)}{ds}\, ds$$

$$= \int_0^L \sum_{n=o}^{\infty} \frac{2n\pi}{L}\left(a_n \cos\frac{2n\pi s}{L} + b_n \sin\frac{2n\pi s}{L}\right) \cdot \left(-c_n \sin\frac{2n\pi s}{L} + d_n \cos\frac{2n\pi s}{L}\right) ds$$

By the orthogonality relations of sine and cosine, this is reduced to:

$$= \frac{2\pi}{L}\ \frac{L}{2} \sum_{n=o}^{\infty} n\left(a_n d_n - b_n c_n\right)$$

$$(3) \quad = \pi \sum_{n=o}^{\infty} n\left(a_n d_n - b_n c_n\right) = A$$

The sign ($\pm$) of the area will depend on the direction in which the curve has been traced.

4.6 Echo Tracing with Pen Plotter

Echo tracing is done internally as a subroutine of FOURAN using the FOUPLO routine. FOUPLO uses the perimeter and the harmonic coefficients. Using relation (1) and (2) in Section 4.5, and terminating the sum at NHAM harmonics, a parametric representation of the echo is obtained. This is easily plotted by breaking up arclength into a suitable number of pieces.

Since the array desired is within the bigger array (See note, Section 4.2), the border will be considered rows 2 and JN-1 and columns 2 and IN-1. No points will be plotted outside this area.

4.7 Advantages and Problems of the Fourier Description

The major advantage of the Fourier method of describing echoes is that all the information about the shape of the echo is represented by the Fourier coefficients generated. The coefficients provide a means of evaluating area, centroid of the contour, and ellipses associated with each echo. The border can also be easily reconstructed in the form of a parametric representation, and plotted.

One problem is that the Fourier representation describes closed contours best. Therefore, in order to close contours actually ending on a boundary, the border of the array was set to a large negative value, which necessitates overdimensioning the array to avoid losing data. Also, a very intricate path may be poorly represented by nine harmonics, although, for the applications described herein, the path description was very satisfactory. (See Blackmer and Duda, 1972.)

A major information loss from the original data occurs because the Fourier coefficients cannot describe data variation within the echo without resorting to multilevel contouring. Although multilevel contouring was experimented with, the Fourier description could not assure that contours would not cross each other when the gradient was steep. The individual contours would be fairly accurate, but they sometimes cross. Another interpolation scheme has been followed in the contouring package.

5. ECHO MATCHING AND TRACKING PROGRAM (TRACK)

5.1 Introduction

The tracking program is divided into two parts, matching and updating. MATCH uses the areas and centroids from two successive sweeps, calculated by the Fourier methods described earlier to provide an echo match between the frames. This information is used to describe echo movement, assign invariant numbers to echoes so they can be more easily followed, and integrate the total rainfall rate following the echo. The program recognizes splits and mergers, lost echoes and new echoes.

The guide to develop this program was intuition. An operator of a radar scope is assumed to be the best echo tracker, so the technique developed tries to duplicate as closely as possible what I have assumed to be his intuitive approach to the tracking problem. Therefore, the procedure used avoids detailed analytic procedures computed on the data, and instead uses only a limited shape description, area and centroid to perform the matching from one frame to the next.

5.2 Calling Sequence for Matching Subroutine

MATCH(AREA1, AREA2, XC1, YC1, XC2, YC2, NECH1, NECH2,
RFALL, K30, FMET1, FMET2, SQ1, SQ2, K3)

AREA1(NECH1) - areas of the echoes in the first frame.

AREA2(NECH2) - areas of the echoes in the second frame.

XC1(NECH1) - X-coordinate centroids in the first frame.

YC1(NECH1) - Y-coordinate centroids in the first frame.

XC2(NECH2) - X-coordinate centroid in second frame.

YC2(NECH2) - Y-coordinate centroid in second frame.

NECH1 - number of echoes in first frame.

NECH2 - number of echoes in second frame.

RFALL(NECH1) - total rainrate of echoes in first frame.

K30 - 30

K3 - 3

FMET1(K30, K3) - The A, B, C metric coefficients discussed in section 5.3 for each old echo.

FMET2(K30, K3) - metric coefficients for the new echoes.

SQ1(NECH1) - the K for each old echo. (See section 5.3)

SQ2(NECH2) - the K for each new echo.

In addition to these call parameters, a COMMON /MAT/ must be used in the program which calls the tracking program. A COMMON /BLOK/ must be used to supply the correct hour and minute.

5.3 Matching Procedure

The essential output from the matching section of MATCH are two arrays, MATCH1 and MATCH2. MATCH1 recognizes splits, while MATCH2 recognizes mergers. MATCH1 (I,K) (K≥2) reters to an echo in frame 2 which has been matched to echo I in frame 1. MATCH2 (I, K) refers to an echo in frame 1 which has been matched to echo I in frame 2. A more detailed interpretation is included in section 5.5.

To generate MATCH1 and MATCH 2, the subroutine proceeds as follows:

a) A predicted displacement from the preceding information available is added to the centroid position of echoes in frame 1. The predicted displacement is an average echo motion of the three preceding time frames for this particular echo. If the echo was found very recently, and three frames are not available, the program uses the previous available times. When an echo is found, initially zero displacement is anticipated. To guard against erratic movement displacing the echo too much, a 36 knot limit for predicted motion is included. Henceforth, p-displacement refers to this predicted displacement.

b) A check for 1-1 matches is made. The best match to echo I with respect to both centroid and area is found and entered into MATCH1 (I, K). If no suitable match is found, this element is left zero. After having attempted matching all echoes in frame 1 to frame 2 by this method, the program attempts to find splits.

c) The program finds and ranks the five best echo matches that have not been matched previously by (b) to other echoes with respect to p-displacement only. Splits are recognized between old echo I1 and new echoes I2 and J2 if the following criteria are satisfied:

(i) The centroid of I2 and J2 is within a close distance of the interior of echo I1.

(ii) The centroid of I2 and J2 taken together is closer to the interior of echo I1 than each echo taken separately.

d) The program then iterates the above procedure with the third best match, but now considers I2 and J2 to be one echo, its centroid and area being the combined centroid and area of I2 and J2. A limit of five fragments for each echo can be located.

e) After all splits have been found, the data from the two frames are reversed, and mergers are found identically with a merger being considered a split backward in time. Mergers are entered into MATCH2.

The foregoing process for ranking echoes and determining whether an echo is a candidate for a split fragment is one which essentially considers only the ellipses matching best to the echo, described by the zeroth and first Fourier coefficient.

Let the best ellipse describing the contour of the echo $\vec{r}(s)$ be $\vec{r}_e(s)$. For simplicity, translate the ellipse to the origin.

$$\vec{r}(s) \approx \vec{r}_e(s) = x(s)\,\hat{i} + y(s)\,\hat{j}$$

$\omega = \frac{2\pi}{L}$ where L is the perimeter

$$\text{a)}\ \vec{r}_e(s) = (a\cos(\omega s) + b\sin(\omega s))\ \hat{i} + (c\ \cos(\omega s) + d\sin\ (\omega s))\ \hat{j}$$

It is also known that an elipse has the general form $x\hat{i} + y\hat{j}$ where the relation between x and y is

$$\text{b)}\ \ Ax^2 + Bxy + Cy^2 = K$$

for some A, B, C and K.

Substituting x and y from a) into b) we see that the relation

$$(Aa^2 + Bac + Cc^2)\ \cos^2(\omega s)$$

$$+ (Ab^2 + Bbd + Cd^2)\ \sin^2\ (\omega s)$$

$$+ (2Aab + B(ad + bc) + 2Ccd)\ \sin(\omega s)\ \cos\ (\omega s) = K$$

Since this must hold for every s, then A,B,C must be restricted by

$$Aa^2 + Bac + Cc^2 = K$$

$$Ab^2 + Bbd + Cd^2 = K$$

$$2\ Aab + B(ad + bc) + 2Ccd = 0$$

Thus there are three equations for A, B and C (given K) which can be solved by matrices.

It will be noticed that the maximum x value will be $\sqrt{a^2 + b^2}$ and the maximum y value will be $\sqrt{c^2 + d^2}$. Thus, the maximum value for $x^2 + y^2$ will be $a^2 + b^2 + c^2 + d^2$ and a reasonable value for K for the bilinear form b) to approximate the standard metric form $x^2 + y^2$ will be $K = (a^2 + b^2 + c^2 + d^2)/2$.

To determine whether the centroid of an echo is inside or near the ellipse, $dis = Adx^2 + Bdxdy + Cdy^2$ where (dx, dy) is the vector between the two centroids is evaluated. If dis is less than K, the centroid is inside the ellipse. If dis is close to K, the point is near the border.

This method can now be applied to determine which new echoes fall within a reasonable distance of the interior of the ellipse. Instead of matching only centroids with respect to distance, the method takes the ellipticity of the echo into consideration. The calculation of the A, B, C, metric coefficients is performed in subroutine METRIC.

Other methods to determine whether a point is inside or outside an echo were investigated, but were deemed too time consuming for the extra information made available.

5.4 Updating Procedure

The update section of MATCH analyzes the arrays MATCH1 and MATCH2 and prints the following information about each echo:

a) Identification number - remains with echo until it is lost, either totally or by merger or split. The first two digits refer to hour found, the second two to minute echo was first found, the third two to the order in which it was found in the frame, and the last digit to the method by which the echo emerged. A last digit of zero means echo was new growth, a "1" means it was the result of a split, and a "2" means it was the result of a merger.

Example:

910051 refers to an echo found at 9:10, fifth in that frame, and was the result of a split.

b) Auxiliary identification number - ranges from 1 to 30 and refers to the index corresponding to the echo. Echo with auxiliary identification of I has volume rainrate (RFALL(I)). This number remains with an echo until the echo is lost, but after that it may be taken by a new echo.

c) Centroid - refers to the (X, Y) position of the echo's centroid. The echo centroid is considered to be the centroid of the curve. This will be quite close to the area centroid except in very unusual cases.

d) Total rainrate - refers to the summed total rainfall rate for the particular time.

e) Total rainfall - the total volume rainfall associated with each echo since the echo was found.

f) Status - self explanatory. NEW refers to next time frame; OLD, LOST refers to preceding time period.

The program finds the status of an echo as follows:

(Note that K ranges between 2 and 6 in the following paragraphs.)

(i) One - one match between I1 and I2 if

MATCH1 (I1, K) = I2, 0, 0, 0, 0 and

MATCH2 (I2, K) = I1, 0, 0, 0, 0

(ii) I1 splits into $(I2_K)$ if

MATCH1 (I1, K) = $I2_1$, $I2_2$, $I2_3$, $I2_4$, $I2_5$.

(iii) $(I1_K)$ merges into I2 if

MATCH2 (I2, K) = $I1_1$, $I1_2$, $I1_3$, $I1_4$, $I1_5$.

(iv) I1 is lost if

MATCH1 (I1, K) = 0, 0, 0, 0, 0 and

MATCH2 (I2, K) $\neq$ I1 for every I2 and K.

(v) I2 is new growth if

MATCH2 (I2, K) = 0, 0, 0, 0, 0 and

MATCH1 (I1, K) $\neq$ I2 for every I1 and K.

g) When the results are plotted, the following shortened description is used. The auxiliary identification is used and the following interpretation is to be made of the letters printed:

NG - new growth

ME - merger

RM - result of merger

SP - split

RS - result of split

LO - lost echo

TR - tracking old echo

5.5 Example of Use of TRACK

The isolation and tracking program was applied to data from June 26, 1973. Because echo activity before 1400 EDT was slight, data was analyzed from 1400 EDT to 2030 EDT. Most of the echo activity occurred after 1730 and the plots from 1730 to 1845 are included at the end of this section.

Approximately 100 echoes were found and tracked between 1400 and 2030, although in reality there was less activity than 100 echoes would indicate. An echo split will be considered one echo lost and two new echoes formed. The tracking program printed 340 descriptions (such as 1905021 tracking), of which I judged approximately ten to be incorrect. Thus, about 97 percent of the computer descriptions were accurate according to my judgment. However, the printout is in such form that the significant echo data may be recovered manually by adding together a couple of numbers in the case of a computer mismatch.

For an example of program output, refer to the next section. A description of echo 1720021 follows. The echoes discussed are labeled with a *.

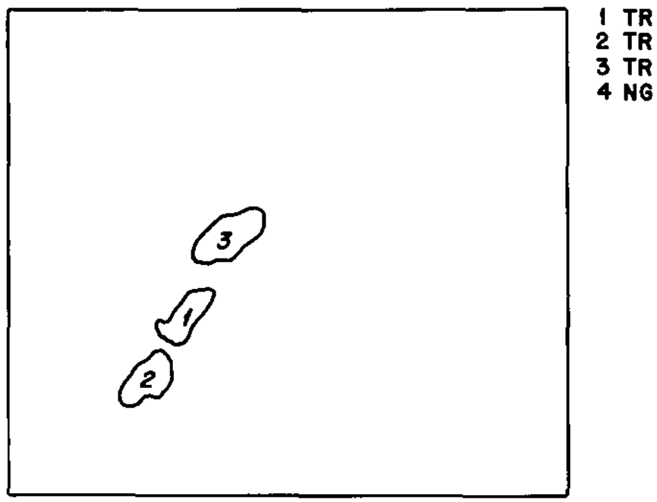
Echo number 1720021 indicates the echo was the result of a split, and was found at 1720. The area of the echo was 37.3 n mi^2 at 1730 and the volume rainrate was 259.0 mm-n mi^2/hr. ($8.89 \times 10^5 m^3$/hr). The total rain volume from this echo since it was found (at 1720) was

31.1 n mi^2-mm (1.07×10^5 m^3). This echo is labeled 3 on the pen plotter picture. At 1730 the echo is heading toward 104.8° at 7.3 knots.

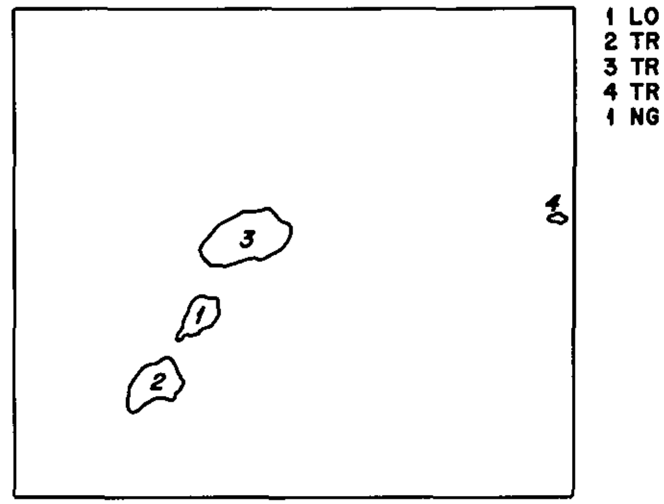
The echo is tracked until 1810 when it merges with echo 1745030 (labeled 5 on the picture) to form a new echo 1815032. So total volume rain fallen from 1720021 was 415.7 mm-n mi^2 (1.4×10^6 m^3) from 1720 until it was lost at 1810.

The new echo 1815032 is large and intense. 1815032 is lost at 1840 after a total rain volume of 464.9 n mi^2-mm (1.6×10^6 m^3) has fallen. The echo splits into echoes 1845041 and 1845081.

5.6 Sample Printouts from TRACK



177 17 30



177 17 35

MAX RADAR ELEV .5 MIN RADAR ELE .4

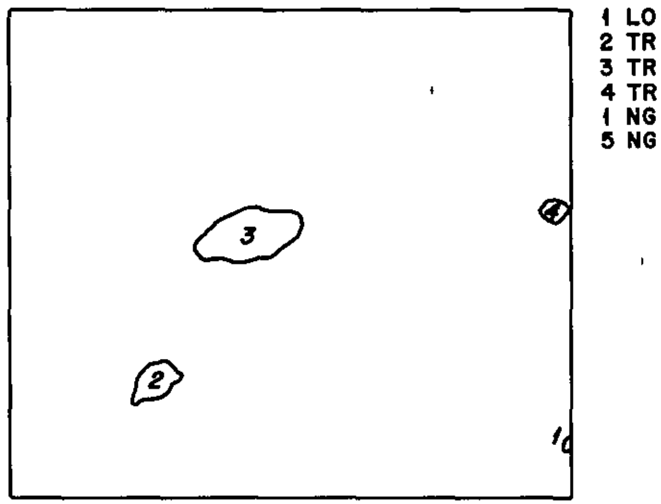
DAY 177.0 HOUR 17.0 MINUTE 30.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT. RN	STATUS
1730021	1	(24.5,24.6)	26.3	69.2	5.8	TRACKING OLD ECHO HEADING 88.3 DEGREES AT 11.4 KNOTS
1730031	2	(19.8,16.5)	29.4	86.2	7.2	TRACKING OLD ECHO HEADING 155.4 DEGREES AT 4.6 KNOTS
*1720021	3	(30.3,34.6)	37.3	259.0	31.1	TRACKING OLD ECHO HEADING 104.8 DEGREES AT 7.3 KNOTS
1735010	4	(71.7,37.0)	2.7			NEW ECHO

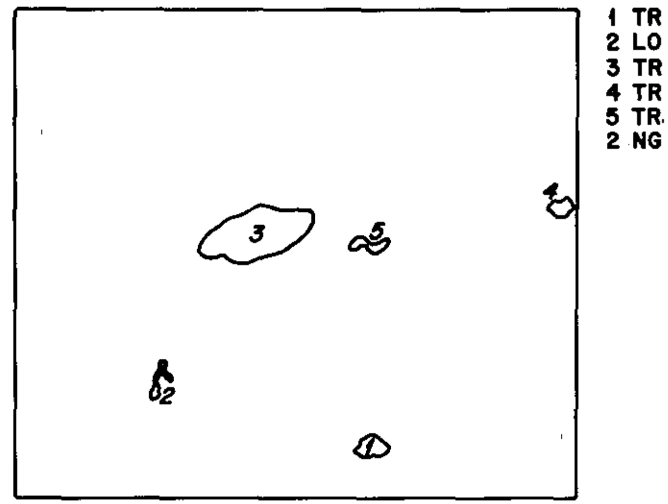
DAY 177.0 HOUR 17.0 MINUTE 35.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT. RN	STATUS
1730021	1	(25.2,24.6)	16.9	29.3	8.2	ECHO LOST
1730031	2	(19.9,16.3)	29.7	95.4	15.1	TRACKING OLD ECHO HEADING 104.3 DEGREES AT 5.3 KNOTS
*1720021	3	(31.6,34.5)	54.8	415.0	65.6	TRACKING OLD ECHO HEADING 101.8 DEGREES AT 6.8 KNOTS
1735010	4	(71.7,37.0)	2.7	6.3	.5	TRACKING OLD ECHO HEADING 22.1 DEGREES AT 8.6 KNOTS
1740010	1	(73.8, 8.8)	2.5			NEW ECHO

* **Means discussed in foregoing section**



177 17 40



177 17 45

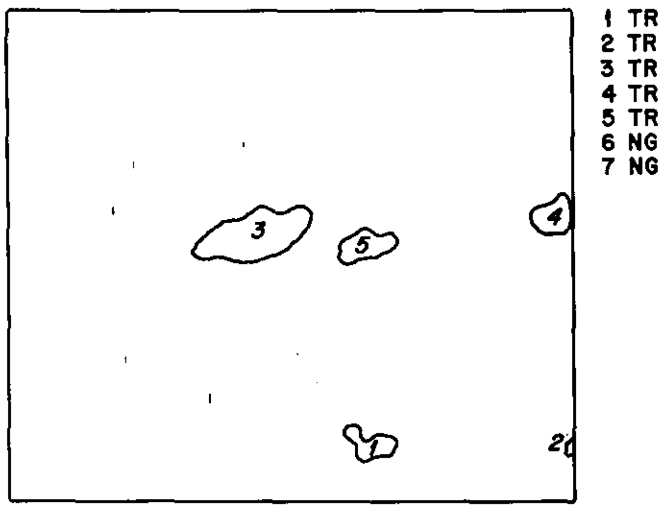
DAY 177.0 HOUR 17.0 MINUTE 40.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1740010	1	(73.8, 8.8)	2.5	11.8	1.0	ECHO LOST
1730031	2	(20.5,16.3)	20.8	47.8	19.1	TRACKING OLD ECHO HEADING 66.6 DEGREES AT 4.2 KNOTS
*1720021	3	(32.5,34.5)	65.4	626.5	117.8	TRACKING OLD ECHO HEADING 86.3 DEGREES AT 6.8 KNOTS
1735010	4	(71.9,37.5)	7.5	41.1	4.0	TRACKING OLD ECHO HEADING 20.9 DEGREES AT 7.0 KNOTS
1745020	1	(48.0, 8.4)	8.3			NEW ECHO
1745030	5	(47.5,33.5)	5.3			NEW ECHO

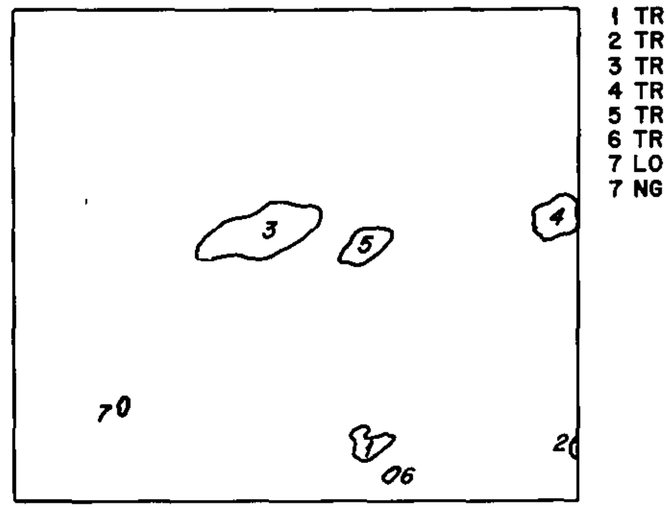
DAY 177.0 HOUR 17.0 MINUTE 45.0

1745020	1	(48.0, 8.4)	8.3	18.6	1.5	TRACKING OLD ECHO HEADING 356.2 DEGREES AT 11.9 KNOTS
1730031	2	(20.7,16.9)	3.3	7.8	19.8	ECHO LOST
*1720021	3	(33.1,34.8)	67.9	718.5	177.7	TRACKING OLD ECHO HEADING 67.9 DEGREES AT 3.6 KNOTS
1735010	4	(72.1,38.0)	5.8	18.7	5.5	TRACKING OLD ECHO HEADING 346.0 DEGREES AT 2.0 KNOTS
1745030	5	(47.5,33.5)	5.3	10.2	.9	TRACKING OLD ECHO HEADING 52.6 DEGREES AT 4.0 KNOTS
1750010	2	(73.8, 8.9)	2.7			NEW ECHO

*Means discussed in foregoing section



177 17 50



177 17 55

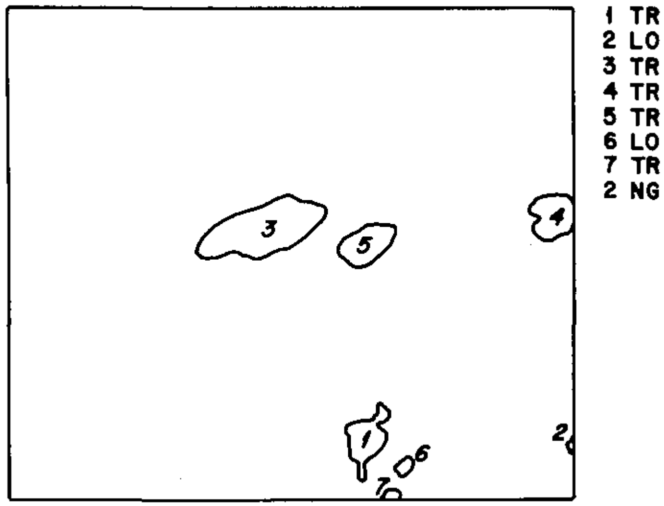
DAY 177.0 HOUR 17.0 MINUTE 50.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1745020	1	(47.9, 9.1)	14.6	45.3	5.3	TRACKING OLD ECHO HEADING 345.3 DEGREES AT 7.4 KNOTS
1750010	2	(73.8, 8.9)	2.7	19.3	1.6	TRACKING OLD ECHO HEADING 145.0 DEGREES AT 3.0 KNOTS
*1720021	3	(33.3,35.1)	67.0	656.9	232.5	TRACKING OLD ECHO HEADING 47.5 DEGREES AT 2.0 KNOTS
1735010	4	(71.6,37.5)	19.5	67.6	11.1	TRACKING OLD ECHO HEADING 238.0 DEGREES AT 2.6 KNOTS
1745030	5	(47.7,33.7)	20.4	54.5	5.4	TRACKING OLD ECHO HEADING 291.2 DEGREES AT 2.3 KNOTS
1755030	6	(50.4, 5.5)	2.4			NEW ECHO
1755070	7	(16.1,13.9)	2.6			NEW ECHO

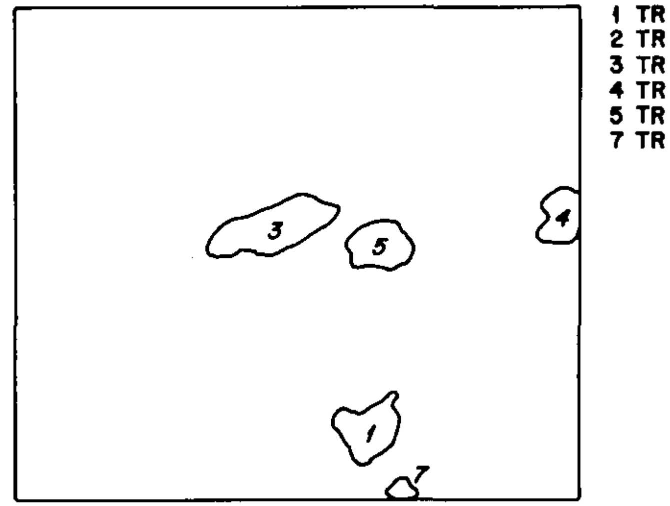
DAY 177.0 HOUR 17.0 MINUTE 55.0

1745020	1	(47.7, 9.4)	12.2	34.3	8.2	TRACKING OLD ECHO HEADING 353.6 DEGREES AT 5.1 KNOTS
1750010	2	(73.9, 8.8)	2.7	13.4	2.7	TRACKING OLD ECHO HEADING 101.2 DEGREES AT 1.0 KNOTS
*1720021	3	(33.5,35.3)	71.5	769.6	296.6	TRACKING OLD ECHO HEADING 51.0 DEGREES AT 1.8 KNOTS
1735010	4	(71.2,37.3)	23.8	148.2	23.5	TRACKING OLD ECHO HEADING 229.5 DEGREES AT 3.2 KNOTS
1745030	5	(47.2,33.7)	19.8	100.5	13.8	TRACKING OLD ECHO HEADING 18.7 DEGREES AT 1.7 KNOTS
1755030	6	(50.4, 5.5)	2.4	4.7	.4	TRACKING OLD ECHO HEADING 69.8 DEGREES AT 26.2 KNOTS
1755070	7	(16.1,13.9)	2.6	5.1	.4	ECHO LOST
1800040	7	(50.7, 2.5)	4.1			NEW ECHO

*Means discussed in foregoing section



177 18 0



177 18 5

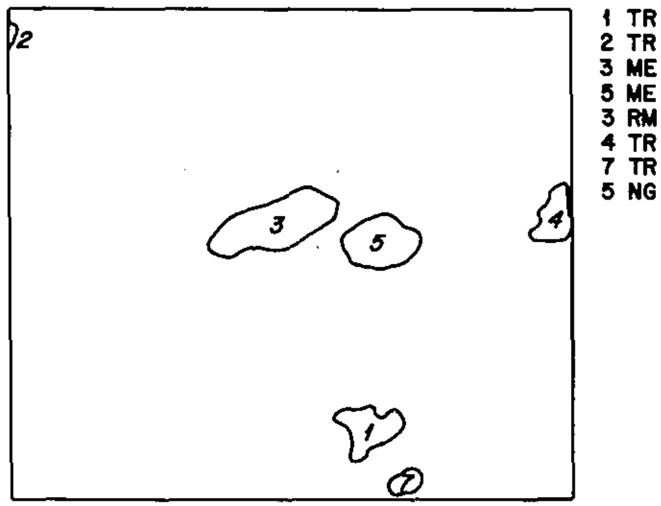
DAY 177.0 HOUR 18.0 MINUTE .0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1745020	1	(47.8, 9.6)	24.6	90.6	15.7	TRACKING OLD ECHO HEADING 346.6 DEGREES AT 5.9 KNOTS
1750010	2	(74.0, 8.9)	2.2	21.0	4.5	ECHO LOST
*1720021	3	(34.0,35.5)	75.9	790.2	362.4	TRACKING OLD ECHO HEADING 63.2 DEGREES AT 2.1 KNOTS
1735010	4	(71.1,37.3)	24.2	267.4	45.8	TRACKING OLD ECHO HEADING 211.9 DEGREES AT .7 KNOTS
1745030	5	(47.7,33.9)	25.8	299.9	38.8	TRACKING OLD ECHO HEADING 112.9 DEGREES AT 2.3 KNOTS
1755030	6	(51.9, 6.0)	4.0	12.3	1.4	ECHO LOST
1800040	7	(50.7, 2.5)	4.1	9.3	.8	TRACKING OLD ECHO HEADING 49.1 DEGREES AT 12.5 KNOTS
1805060	2	(2.1,60.6)	2.1			NEW ECHO

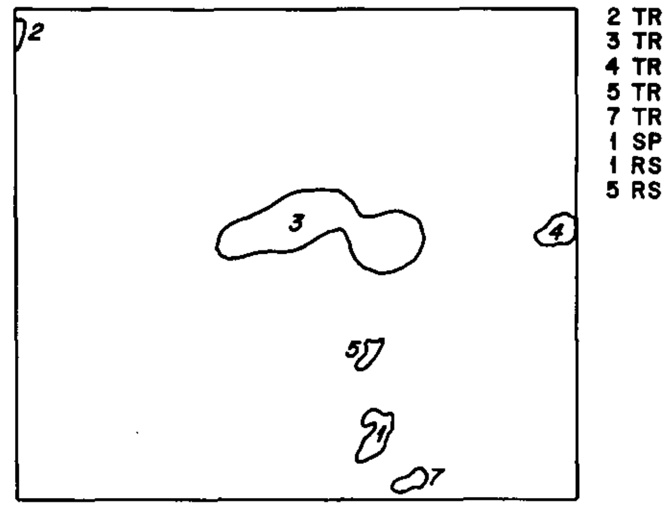
DAY 177.0 HOUR 18.0 MINUTE 5.0

1745020	1	(47.4,11.2)	44.9	146.5	27.9	TRACKING OLD ECHO HEADING 358.6 DEGREES AT 3.0 KNOTS
1805060	2	(2.1,60.6)	2.1	4.3	.4	TRACKING OLD ECHO HEADING 170.4 DEGREES AT 7.6 KNOTS
*1720021	3	(34.8,35.8)	74.2	638.9	415.7	TRACKING OLD ECHO HEADING 67.9 DEGREES AT 2.5 KNOTS
1735010	4	(71.4,37.2)	28.8	228.2	64.8	TRACKING OLD ECHO HEADING 101.6 DEGREES AT .7 KNOTS
1745030	5	(48.4,33.5)	40.6	233.7	58.2	TRACKING OLD ECHO HEADING 82.0 DEGREES AT 5.2 KNOTS
1800040	7	(51.3, 3.0)	9.3	38.6	4.0	TRACKING OLD ECHO HEADING 46.5 DEGREES AT 14.1 KNOTS

*Means discussed in foregoing section



177 18 10



177 18 15

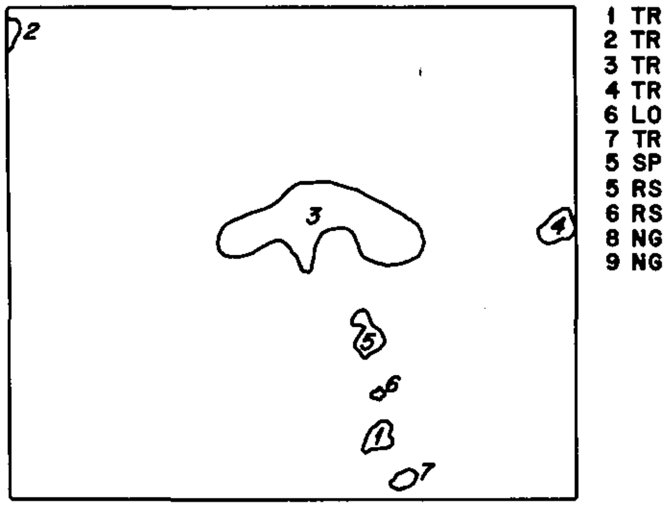
DAY 177.0 HOUR 18.0 MINUTE 10.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1745020	1	(47.6,10.8)	33.8	89.7	35.4	TRACKING OLD ECHO HEADING 12.2 DEGREES AT 1.9 KNOTS
1805060	2	(2.2,60.1)	4.2	8.9	1.1	TRACKING OLD ECHO HEADING 172.7 DEGREES AT 4.6 KNOTS
*1720021	3	(35.4,36.1)	80.0	661.7	.0	LOST MERGED
*1745030	5	(49.2,34.0)	49.7	390.4	.0	LOST MERGED
*1815032	3	(41.7,35.6)	130.9			NEW RESULT OF MERGER
1735010	4	(71.6,37.1)	25.5	110.8	74.0	TRACKING OLD ECHO HEADING 168.0 DEGREES AT 2.4 KNOTS
1800040	7	(52.2, 3.9)	9.3	44.2	7.7	TRACKING OLD ECHO HEADING 43.0 DEGREES AT 11.3 KNOTS
1815050	5	(47.4,20.6)	6.5			NEW ECHO

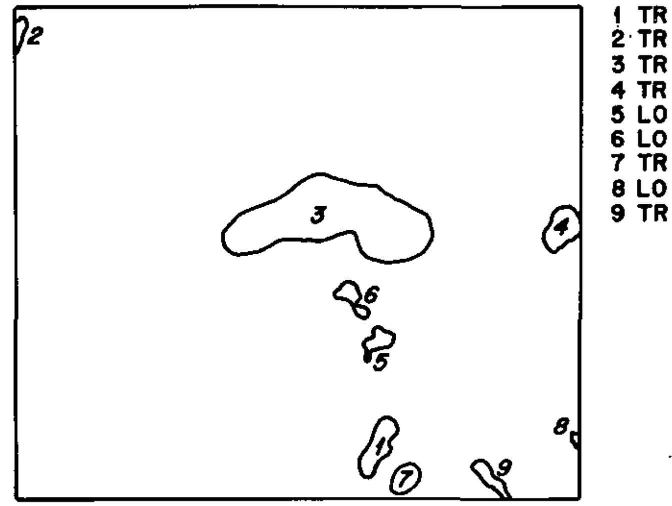
DAY 177.0 HOUR 18.0 MINUTE 15.0

1805060	2	(2.2,60.0)	5.8	13.2	2.4	TRACKING OLD ECHO HEADING 168.3 DEGREES AT 4.2 KNOTS
*1815032	3	(41.7,35.6)	130.9	1201.1	120.1	TRACKING OLD ECHO HEADING 170.6 DEGREES AT 4.7 KNOTS
1735010	4	(71.4,35.7)	15.5	82.3	82.3	TRACKING OLD ECHO HEADING 169.1 DEGREES AT 2.1 KNOTS
1815050	5	(47.4,20.6)	6.5	13.6	1.4	TRACKING OLD ECHO HEADING 355.4 DEGREES AT 18.5 KNOTS
1800040	7	(52.5, 4.4)	7.8	18.9	9.6	TRACKING OLD ECHO HEADING 21.5 DEGREES AT 3.8 KNOTS
1745020	1	(48.0,10.6)	16.0	47.3	40.1	LOST, SPLIT
1821041	1	(48.7, 9.6)	10.1			NEW RESULT OF SPLIT
1821051	6	(48.8,14.9)	2.1			NEW RESULT OF SPLIT

*Means discussed in foregoing section



177 18 21



177 18 25

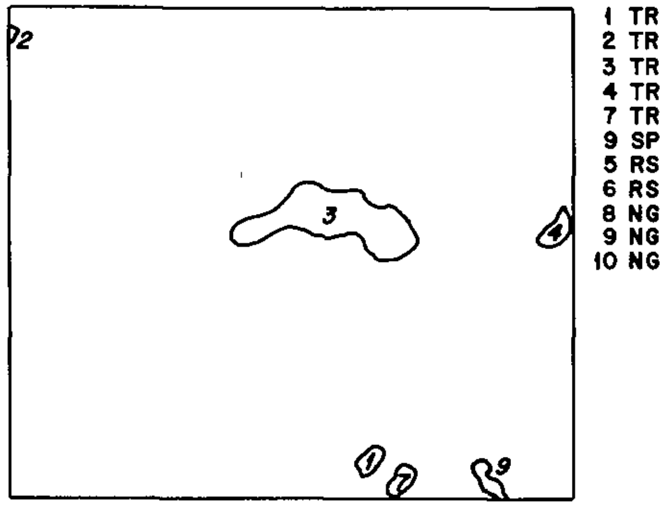
DAY 177.0 HOUR 18.0 MINUTE 21.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1821041	1	(48.7, 9.6)	10.1	23.0	1.5	TRACKING OLD ECHO HEADING 195.4 DEGREES AT 21.1 KNOTS
1805060	2	(2.3,59.5)	6.7	14.3	3.4	TRACKING OLD ECHO HEADING 163.0 DEGREES AT 1.5 KNOTS
*1815032	3	(41.7,35.3)	142.8	810.2	174.1	TRACKING OLD ECHO HEADING 47.1 DEGREES AT 4.7 KNOTS
1735010	4	(71.7,35.6)	14.0	84.9	87.9	TRACKING OLD ECHO HEADING 181.1 DEGREES AT 2.0 KNOTS
1821051	6	(48.8,14.9)	2.1	3.5	.2	ECHO LOST
1800040	7	(52.0, 4.4)	6.1	16.7	10.7	TRACKING OLD ECHO HEADING 293.0 DEGREES AT 1.4 KNOTS
1815050	5	(47.3,22.1)	12.7	27.1	3.2	LOST, SPLIT
1825071	5	(48.2,21.1)	7.6			NEW RESULT OF SPLIT
1825081	6	(45.1,26.4)	7.6			NEW RESULT OF SPLIT
1825010	8	(73.8, 9.2)	2.8			NEW ECHO
1825030	9	(62.8, 4.4)	8.5			

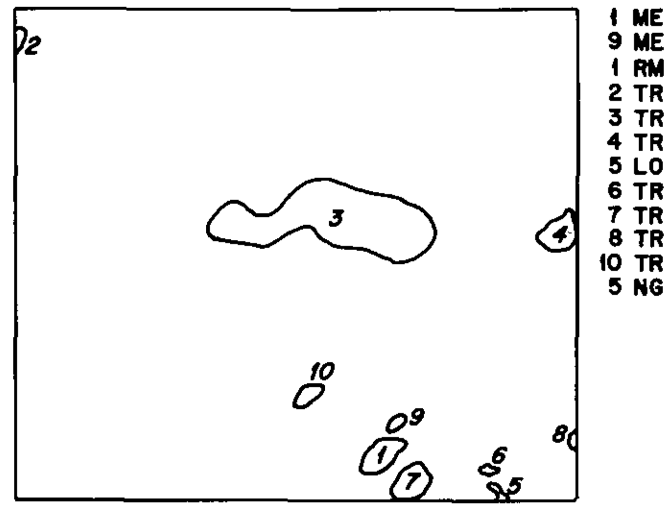
DAY 177.0 HOUR 18.0 MINUTE 25.0

1821041	1	(48.5, 8.7)	18.3	60.0	6.5	TRACKING OLD ECHO HEADING 194.7 DEGREES AT 19.2 KNOTS
1805060	2	(2.4,59.5)	5.8	12.5	4.4	TRACKING OLD ECHO HEADING 175.9 DEGREES AT .9 KNOTS
*1815032	3	(42.2,36.1)	161.9	1102.0	266.0	TRACKING OLD ECHO HEADING 61.7 DEGREES AT 5.4 KNOTS
1735010	4	(71.6,35.5)	17.5	220.5	106.3	TRACKING OLD ECHO HEADING 118.5 DEGREES AT .6 KNOTS
1825071	5	(48.2,21.1)	7.6	14.4	1.2	ECHO LOST
1825081	6	(45.1,26.4)	7.6	14.6	1.2	ECHO LOST
1800040	7	(51.9, 4.3)	10.7	44.0	14.4	TRACKING OLD ECHO HEADING 253.7 DEGREES AT 1.4 KNOTS
1825010	8	(73.8, 9.2)	2.8	22.0	1.8	ECHO LOST
1825030	9	(62.8, 4.4)	8.5	24.2	2.0	TRACKING OLD ECHO HEADING 111.2 DEGREES AT 6.6 KNOTS

*Means discussed in foregoing section



177 18 30



177 18 35

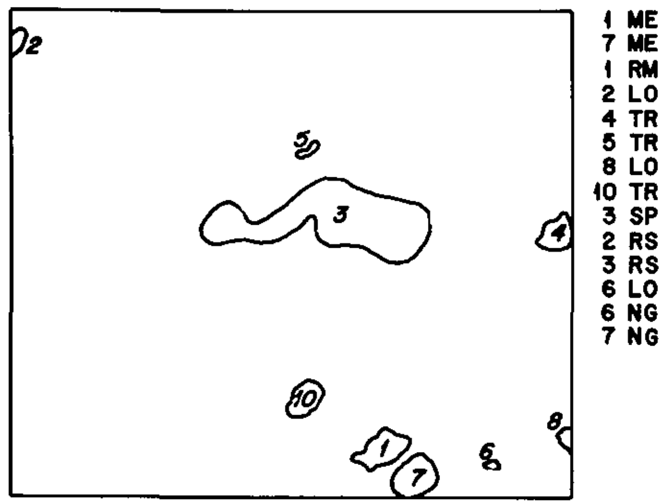
DAY 177.0 HOUR 18.0 MINUTE 30.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1821041	1	(48.1, 7.2)	8.9	25.6	8.7	TRACKING OLD ECHO HEADING 164.7 DEGREES AT 8.8 KNOTS
1805060	2	(2.2,59.5)	3.3	7.1	5.0	TRACKING OLD ECHO HEADING 184.9 DEGREES AT 1.2 KNOTS
*1815032	3	(42.8,36.2)	109.0	338.9	294.2	TRACKING OLD ECHO HEADING 336.5 DEGREES AT 4.2 KNOTS
1735010	4	(71.9,35.4)	12.2	52.5	110.7	TRACKING OLD ECHO HEADING 160.1 DEGREES AT .3 KNOTS
1800040	7	(51.9, 4.3)	8.9	23.2	16.3	TRACKING OLD ECHO HEADING 100.0 DEGREES AT 1.8 KNOTS
1825030	9	(63.2, 4.3)	9.7	24.4	4.1	LOST, SPLIT
1835031	5	(64.3, 2.8)	3.2			NEW RESULT OF SPLIT
1835041	6	(63.0, 6.0)	2.3			NEW RESULT OF SPLIT
1835010	8	(74.0, 9.5)	3.7			NEW ECHO
1835080	9	(51.0,11.6)	3.8			NEW ECHO
1835090	10	(39.6,15.0)	7.8			NEW ECHO

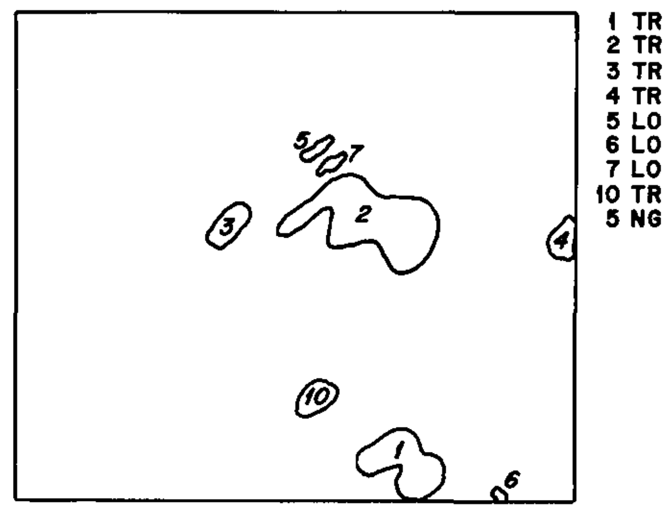
DAY 177.0 HOUR 18.0 MINUTE 35.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1821041	1	(49.2, 7.7)	17.2	120.2	.0	LOST MERGED
1835080	9	(51.0,11.6)	3.8	6.5	.0	LOST MERGED
1840062	1	(49.6, 8.0)	22.1			NEW RESULT OF MERGER
1805060	2	(2.3,58.9)	4.3	8.9	5.8	TRACKING OLD ECHO HEADING 159.4 DEGREES AT 1.6 KNOTS
*1815032	3	(41.2,36.4)	158.0	981.9	376.0	TRACKING OLD ECHO HEADING 287.6 DEGREES AT 2.8 KNOTS
1735010	4	(71.8,35.3)	17.5	184.7	126.1	TRACKING OLD ECHO HEADING 136.6 DEGREES AT .5 KNOTS
1835031	5	(64.3, 2.8)	3.2	9.3	.8	ECHO LOST
1835041	6	(63.0, 6.0)	2.3	6.1	.5	TRACKING OLD ECHO HEADING 88.8 DEGREES AT 8.5 KNOTS
1800040	7	(53.0, 4.2)	17.3	119.9	26.3	TRACKING OLD ECHO HEADING 81.3 DEGREES AT 3.0 KNOTS
1835010	8	(74.0, 9.5)	3.7	32.5	2.7	TRACKING OLD ECHO HEADING 268.1 DEGREES AT 4.2 KNOTS
1835090	10	(39.6,15.0)	7.8	15.2	1.3	TRACKING OLD ECHO HEADING 163.4 DEGREES AT 7.9 KNOTS
1840080	5	(40.1,45.5)	2.5			NEW ECHO

*Means discussed in foregoing section



177 18 40



177 18 45

DAY 177.0 HOUR 18.0 MINUTE 40.0

IDENTIFICATION		CENTROID	AREA	TOT.RT	TOT.RN	STATUS
1840062	1	(49.6, 8.0)	22.1	145.4	.0	LOST MERGED
1800040	7	(53.8, 4.6)	23.5	227.7	.0	LOST MERGED
1845032	1	(51.9, 6.6)	50.8			NEW RESULT OF MERGER
1805060	2	(2.7,58.6)	7.1	18.2	7.3	ECHO LOST
1735010	4	(72.0,35.0)	15.2	134.6	137.3	TRACKING OLD ECHO HEADING 145.4 DEGREES AT .6 KNOTS
1840080	5	(40.1,45.5)	2.5	6.1	.5	TRACKING OLD ECHO HEADING 51.6 DEGREES AT 6.7 KNOTS
1835010	8	(73.7, 9.5)	5.5	22.2	4.6	ECHO LOST
1835090	10	(39.8,14.5)	14.8	69.6	7.1	TRACKING OLD ECHO HEADING 103.0 DEGREES AT 6.8 KNOTS
*1815032	3	(41.2,36.5)	133.9	1066.3	464.9	LOST, SPLIT
*1845041	2	(46.0,36.8)	111.1			NEW RESULT OF SPLIT
*1845081	3	(29.3,36.4)	18.0			NEW RESULT OF SPLIT
1835041	6	(63.6, 6.0)	2.5	9.4	1.3	ECHO LOST
1845020	6	(64.5, 2.8)	2.6			NEW ECHO
1845050	7	(42.5,44.0)	4.5			NEW ECHO

*Means discussed in foregoing section

6. ACKNOWLEDGMENTS

I am very grateful for the assistance Victor Wiggert has given. He suggested this computer project, and was helpful with suggestions and programming hints, as well as with the writing of this paper. I would also like to thank Dr. William Woodley for many helpful suggestions and for his interest in my project.

7. REFERENCES

Blackmer, R. H. and R. O. Duda, 1972: Application of Pattern Recognition Techniques to Digitized Radar Data. Stanford Research Institute, Contract 1 - 36072 Report.

Booth A. L., A. G. De Cotiis, and J. A. Leese, 1972: Automated Procedures for Mapping and Display of Digitized Radar Data. Preprint Volume of the 15th Radar Meteorology Conference, 1972. 199-203.

Barclay, P. A. and K. A. Wilk, 1970: Severe Thunderstorm Radar Echo Motion and Related Weather Events Hazardous to Aviation Operators ESSA Technical Memorandum ERL TM-NSSL 46.

Wiggert, V., and G.F. Andrews, 1974: Digitizing, Recording and Computer Processing Weather Radar Data at EML. To appear as NOAA Technical Memorandum.

Appendix A - KART

(i) Main Program Calling KART

```
FOR,SI    ,.MAIN
      DIMENSION RESP(73,62)
      COMMON /BLOK/IBINS,IN ,JN ,DIS,PI,XZ,YZ,TDIFF,TN1,BN1(300),
     1BN2(300),KCART,PHI1,AZM1,TDA1,TH1,TM1,ELMAX,ELMIN ,TE,TDA,TH,TM
      REWIND 2
      REWIND 3
      KCART=0
      KEY=1
      NECH=30
      SMT=540.
      EMT=1330
      IN=73
      IM=IN
      JN=62
      JM=JN
      PI=3.1415923
      DIS=1.
      IBINS=300
      XZ=-74.695
      YZ=13.405
      TN1=0.
      DO 10 IJK=1,180
      CALL KART(RESP,SMT,EMT,KEY)
      CALL MAPP(RESP,IM,JM)
      DO 5 I=1,IN
    5 WRITE(3)(RESP(I,J),J=1,JN)
      WRITE(3)TDIFF,TDA1,TH1,TM1,ELMAX,ELMIN
    6 CONTINUE
   10 CONTINUE
      END FILE 3
      REWIND 2
      REWIND 3
      CALL EXIT
      END
```

(ii) Subroutine KART

```
[FOR,SI   ,.SUB5
      SUBROUTINE KART(RESP,SMT,EMT,KEY)
      DIMENSION RESP(IN,JN)
      COMMON /BLOK/IBINS,IN ,JN ,DIS,PI,XZ,YZ,TDIFF,TN1,BN1(300),
     1BN2(300),KCART,PHI1,AZM1,TDA1,TH1,TM1,ELMAX,ELMIN ,TE,TDA,TH,TM
      KCART=KCART+1
      KEY=0
      ELMIN=100.
      ELMAX=-100.
      KNTC=0
      RACON=1.706E-12
      PQ = -10.*ALOG10(RACON*300.)
      DO 11 I=1,IN
      DO 11 J=1,JN
   11 RESP(I,J)=0.
      IF(KCART.NE.1)GO TO 3
   20 CONTINUE
   15 CONTINUE
      READ(2                         )IL,IC,TDE,TDA,TH,TM,TS,AZM1,TE,TRI,TRD,
     1  IK,IBG,IEM,(BN1(K),K=1,IBINS),TB,TER
      IF(AZM1.LT.270.3)GO TO 15
      TN1=TH+TM/60.
      TIME=60*TH+TM
      IF(TIME.LT.SMT)GO TO 20
      DO 805 K=1,IBINS
  805 BN1(K)=10.**((BN1(K)+PQ)/14.)
      PHI1=(90.-AZM1)*PI/180.
    3 CONTINUE
      TDA1=TDA
      TH1=TH
      TM1=TM
      IF(TE.GT.1.01.AND.TE.LT.23.0)KEY=2
      IF(TE.GT.ELMAX)ELMAX=TE
      IF(TE.LT.ELMIN)ELMIN=TE
      TN2=TH+TM/60.
   14 CONTINUE
      READ(2                         )IL,IC,TDE,TDA,TH,TM,TS,AZM2,TE,TRI,TRD,
     1  IK,IBG,IEM,(BN2(K),K=1,IBINS),TB,TER
      IF(AZM2.LT.270.3)GO TO 14
      TIME=60.*TH+TM
      IF(TIME.GT.EMT)KEY=1
      IF(ABS(AZM1-AZM2).GT.30.)GO TO 150
      GTLN=TRI
      PHI2=(90.-AZM2)*PI/180.
```

```
      COT2=COTAN(PHI2)
      COT1=COTAN(PHI1)
      DO 905 K=1,IBINS
  905 BN2(K)=10.**((BN2(K)+PQ)/14.)
      DO 5 J=1,JN
      IF(PHI1.LT.-PI)PHI1=PHI1+2.*PI
      IF(PHI2.LT.-PI)PHI2=PHI2+2.*PI
      I1=INT(((YZ+J*DIS)*COT2          -XZ)/DIS)
      I2=INT(((YZ+J*DIS)*COT1          -XZ)/DIS)
      IE=MAX0(I1,I2)
      IB=MIN0(I1,I2)
      IF(IB.EQ.IE)GO TO 5
      IB=IB+1
      IF(IB.LT.1)IB=1
      IF(IB.GT.IN)GO TO 5
      IF(IE.GT.IN)IE=IN
      IF(IE.LT.1)GO TO 5
      Y=YZ+J*DIS
      DO 4 I=IB,IE
      X=XZ+I*DIS
      R=SQRT(X**2+Y**2)
      PHI=ATAN2(Y,X)
      S=ABS((PHI-PHI1)/(PHI2-PHI1) )
      R1=INT(R/GTLN)*GTLN
      R2=R1+GTLN
      T=(R-R1)/(R2-R1)
      IBN1=INT(R/GTLN)
      IBN2=IBN1+1
      F=BN1(IBN2)+S*(BN2(IBN2)-BN1(IBN2))
      G=BN1(IBN1)+S*(BN2(IBN1)-BN1(IBN1))
    4 RESP(I,J)=G+T*(F-G)
    5 CONTINUE
      AZM1=AZM2
      PHI1=PHI2
      DO 6 K=1,IBINS
    6 BN1(K)=BN2(K)
      GO TO 3
  150 CONTINUE
      TDIFF=TN2-TN1
      TN1=TN2
      AZM1=AZM2
      RETURN
      END
```

(iii) Subroutine MAPP

```
(FOR,SI   ,.SUB9
      SUBROUTINE MAPP(Z,IE,JE)
      COMMON /BLOK/IBINS,IN ,JN ,DIS,PI,XZ,YZ,TDIFF,TN1,BN1(300),
     1BN2(300),KCART,PHI1,AZM1,TDA1,TH1,TM1,ELMAX,ELMIN ,TE,TDA,TH,TM
      DIMENSION Z(IE,JE),CHAR(13),PR( 131)
      DATA CHAR /1H ,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H+,1HB,1H./
      FMIN=Z(1,1)
      FMAX=Z(1,1)
      DO 1 J=1,JE
      DO 1 I=1,IE
      IF(FMIN.GT.Z(I,J))FMIN=Z(I,J)
    1 IF(FMAX.LT.Z(I,J))FMAX=Z(I,J)
      FDIFF=FMAX-FMIN
   12 WRITE(6,3)FMIN,FMAX,FDIFF
    9 SF=10./FDIFF
      RSF=1./SF
      WRITE(6,2)RSF,FMIN
    2 FORMAT(1H0,10X,11HTRUE VALUE=,F11.5,17H X PRINT VALUE + ,F11.5)
    3 FORMAT(1H1,10X,14HMINIMUM VALUE=,E12.5,5X,14HMAXIMUM VALUE=,E12.5,
     15X,10HDIFFRENCE=,E12.5)
      JE2=2*JE+2
      DO 5 K=1,JE2
    5 PR(K)=CHAR(12)
      WRITE(6,99)(PR(K),K=1,JE2)
      JE1=JE+1
      DO 10 I=1,IE
      DO 8 K=2,JE1
      IP=INT((Z(I,K-1)-FMIN)*SF)+1
      IF(IP.EQ.1.AND.Z(I,K-1).GT.1.)IP=13
      KE=2*K   -2
      KO=KE+1
      PR(KO)=CHAR(1)
      IF(MOD(I,10).NE.0)GO TO 8
      DO 7 ID=21,JE2,20
    7 PR(ID)=CHAR(11)
    8 PR(KE)=CHAR(IP)
      WRITE(6,99)(PR(K),K=1,JE2)
   10 CONTINUE
      DO 11 K=1,JE2
   11 PR(K)=CHAR(12)
      WRITE(6,99)(PR(K),K=1,JE2)
   99 FORMAT(1H ,131A1)
      WRITE(6,97)ELMAX,ELMIN
   97 FORMAT(1H0,10X,14HMAX RADAR ELEV,F5.1,10X,14H MIN RADAR ELE,F5.1)
      WRITE(6,98)TDA1,TH1,TM1
   98 FORMAT(1H0,10X,3HDAY,F6.1,10X,4HHOUR,F5.1,10X,6HMINUTE,F5.1)
      RETURN
      END
```

Appendix B - Program RSUM

(i) Main Program

```
[FOR,SI   ,.MAIN
      COMMON/PLO/IBUF(1024)
      COMMON/BLOCK/IB(6),IE(6),JB(6),JE(6),RGA1(6),RGA2(6),RESP(73,62),
     1     TIME,TDA1,IDO  ,TDIFF,BEGM(6,5),ENDM(6,5)
      DIMENSION ID(6),RINT(73,62),FIM(6,6),SOU(7,5),CYP(6,4),
     1     PAH(6,7),GAR(21,13)
      DATA/RINT,FIM,SOU,CYP,PAH,GAR/4936*0./
C                    IB IS THE BEGINNING COLUMN OF THE CLUSTERS
C                    IE IS THE ENDING COLUMN OF THE CLUSTERS
C     JE IS THE ENDING ROW
C     JB IS THE BEGINNING ROW
C      NOTE THAT I DIMENSIONS REFER TO X COORDINATE , J REFERS TO Y
C     IN,JN       ARE  THE I AND J DIMENSIONS OF THE ENTIRE TARGET
      DATA/IB/0,11,41,36,53,43/
      DATA/IE/72,16,47,41,58,63/
      DATA/JB/0,24,39,18,47,27/
      DATA/JE/61,29,43,21,53,39/
      DATA/RGA1/6HENTIRE,6HIMOKAL,6HSOUTH ,6HBIG CY,6HPAHOKE,6HGARSTA /
      DATA/RGA2/6H TARG.,6HLEE   ,6HBAY   ,6HPRESS ,6HE     ,6HNG NET /
      CALL PLOTS(IBUF,1024,9)
      IN=73
      JN=62
      ET=0.
C     READ IN THE STARTING AND ENDING TIMES
      DO 4 I=1,6
      READ(5,2)(BEGM(I,K),ENDM(I,K)  ,K=1,5 )
    2 FORMAT(5(1X,F5.2,1X,F5.2))
C     FIND THE LATEST ENDING TIME
      DO 1 K=1,5
    1 IF(ENDM(I,K).GT..01.AND.ET.LT.ENDM(I,K))ET=ENDM(I,K)
C     ID(K) REFERS TO THE TIME PERIOD OF CLUSTER K BEING SUMMED OVER
    4 ID(I)=1
      TIMO=BEGM(1,1)
      TIMS=TIMO
      KTM=0
C     KTM IS THE NUMBER OF TIMES THIS SECTION HAS BEEN RUN
    5 CONTINUE
      KTM=KTM+1
```

```
C       READ IN DATA
      DO 6 I=1,IN
    6 READ(3)(RESP(I,J),J=1,JN)
      READ(3)TDIFF,TDA1,TH1,TM1,ELMAX,ELMIN
      IF(KTM.EQ.1)GO TO 5
      TIME=TH1+.01*TM1
      DO 50 K=1,6
      IDO=ID(K)
      IF(BEGM(K,IDO).LT..01)GO TO 50
      IF(BEGM(K,IDO).GT.TIME)GO TO 50
C       PERFORM THE INTEGRATION OVER THE LATEST TIME PERIOD
      IF(K.EQ.1)CALL SUM(RINT, IE(K)-IB(K)+1,JE(K)-JB(K)+1 ,K)
      IF(K.EQ.2)CALL SUM(FIM , IE(K)-IB(K)+1,JE(K)-JB(K)+1 ,K)
      IF(K.EQ.3)CALL SUM(SOU , IE(K)-IB(K)+1,JE(K)-JB(K)+1 ,K)
      IF(K.EQ.4)CALL SUM(CYP , IE(K)-IB(K)+1,JE(K)-JB(K)+1 ,K)
      IF(K.EQ.5)CALL SUM(PAH , IE(K)-IB(K)+1,JE(K)-JB(K)+1 ,K)
      IF(K.EQ.6)CALL SUM(GAR , IE(K)-IB(K)+1,JE(K)-JB(K)+1 ,K)
      IF(ENDM(K,IDO).GT.TIME)GO TO 50
      ID(K)=ID(K)+1
      IF(ET.LT.TIME-.1)GO TO 99
   50 CONTINUE
C       EVERY HOUR PRINT THE TOTAL TARGET
      IF(TIME.LT.TIMO)GO TO 5
      CALL MAPP(RINT,IN,JN)
      SUMA=0.
      DO 10 I=1,IN
      JS=INT(.288*(I-1)+41.4)
      DO 10 J=1,JS
   10 SUMA=SUMA+RINT(I,J)
      SUMA=SUMA*3434.27
      WRITE(6,11)TIMS,TIMO,TDA1
   11 FORMAT(1H0,10X,20H ENTIRE TARGET FROM  ,F5.2,3H TO ,F5.2,4H DAY,
     1     F4.0)
      WRITE(6,12)SUMA
   12 FORMAT(1H0,10X,15HTOTAL RAINFALL  ,E12.5,13H CUBIC METERS  )
      TIMO=TIMO+1.
      GO TO 5
   99 CONTINUE
      CALL EXIT
      END
```

(ii) Subroutine SUM

```
[FOR,SI     ,.SUBC
      SUBROUTINE SUM(Z,IEN,JEN,K)
      COMMON/PLO/IBUF(1024)
      COMMON/BLOCK/IB(6),IE(6),JB(6),JE(6),RGA1(6),RGA2(6),RESP(73,62),
     1      TIME,TDA1,IDO ,TDIFF,BEGM(6,5),ENDM(6,5)
      DIMENSION Z(IEN,JEN)
      IEN1=IEN-1
      JEN1=JEN-1
      SPX=.5
      SPY=.5
C     FIND THE RIGHT SCALE TO BE USED IN PLOTTING
      IF(K.EQ.1)SPX=.1
      IF(K.EQ.1)SPY=.1
      DO 1 I=1,IEN
      DO 1 J=1,JEN
      IDEX=I   +IB(K)
      JDEX=J   +JB(K)
    1 Z(I,J)=Z(I,J)+TDIFF*RESP(IDEX,JDEX)
      IF(ENDM(K,IDO).GT.TIME)RETURN
C     THE NEW ADDITION TO THE TIME INTEGRATION
      WRITE(6,15)
   15 FORMAT(1H1)
      IF(K.NE.1)GO TO 2
      CALL MAPP(Z,IEN,JEN)
C     FIND TOTAL TARGET RAINFALL IN CUBIC METERS
      SUMA=0.
      DO 11 I=1,IEN
      JS=INT(.2887*(I-1)+41.4)
      DO 11 J=1,JS
   11 SUMA=SUMA+Z(I,J)
      SUMA=SUMA*3434.27
      WRITE(6,12)SUMA
   12 FORMAT(1H0,10X,31HTOTAL TRAPEZOID TARGET RAINFALL    ,
     1      E12.5,13H CUBIC METERS   )
      GO TO 3
```

```
    2 CONTINUE
      IF(K.NE.1)WRITE(6,16)
   16 FORMAT(1H0,20X,5HNORTH)
      DO 6 J=1,JEN
      JR=JEN-J+1
    6 WRITE(6,7)(Z(I,JR),I=1,IEN)
    7 FORMAT(1H0,/,10X,24F5.1)
    3 WRITE(6,8)RGA1(K),RGA2(K),BEGM(K,IDO),ENDM(K,IDO),TDA1
    8 FORMAT(1H0,10X,2A6,8H BETWEEN ,F6.2,4H AND,F6.2, 4H DAY,F4.0,
     1      21H TOTAL RAINFALL IN MM,//    )
      AMI=Z(1,1)
      AMA=Z(1,1)
      DO 13 I=1,IEN
      DO 13 J=1,JEN
      IF(Z(I,J).LT.AMI)AMI=Z(I,J)
      IF(Z(I,J).GT.AMA)AMA=Z(I,J)
   13 CONTINUE
      CINTER=(AMA-AMI)/6.
C       CONTOUR THE SUM ARRAY
      CALL CLOT(Z,IEN,JEN,CINTER,1.,SPX,SPY,1)
C         REINITIALIZE TO  ZERO
      DO 10 I=1,IEN
      DO 10 J=1,JEN
   10 Z(I,J)=0.
      RETURN
      END
```

Appendix C - Tracking and Fourier Analysis Program

Main Program

```
[FOR,SI     ,.MAIN
      DIMENSION COSX(30),SINX(30),CROS(30)
      COMMON/MAT/MATCH1(30,6),MATCH2(30,6),NUM(30),IETN(30),IDENT(30),
     1 RTOT(30),MERGN(6 ),MERGI(6 ),NSPL(6)
     2         ,XL(3,30),YL(3,30),TL(3,30)
      DIMENSION FMET1(3,30),FMET2(3,30)
      DIMENSION AREA1(30),AREA2(30),XC1(30),XC2(30),YC1(30),YC2(30)
      DIMENSION SQ1(30),SQ2(30)
      DIMENSION AREA(30),RFAL1(30),RFAL2(30)
      DIMENSION COF(9,4,30),PERIM(30),RFALL(30),IBRDS(30)
      COMMON/PLO/IBUF(1024)
      DIMENSION RESP(75,64)
      COMMON /BLOK/IBINS,IN ,JN ,DIS,PI,XZ,YZ,TDIFF,TN1,BN1(300),
     1BN2(300),KCART,PHI1,AZM1,TDA1,TH1,TM1,ELMAX,ELMIN ,TE,TDA,TH,TM
      CALL PLOTS(IBUF,1024,9)
      CALL FACTOR(.07)
      SMT=540
      EMT=680
      THRESH=1.
      NHAM=9
      NECH=30
      KEY=1
      KCART=0
      REWIND 2
      PI=3.14159265
      DIS=1.
      IN=75
      JN=64
      IN1=IN-1
      JN1=JN-1
      IM=IN
      JM=JN
      DO 33 I1=1   ,30
      RTOT(I1)=0
      NUM(I1)=0
      IDENT(I1)=0
       IETN(I1)=0
   33 CONTINUE
      IBINS=300
      XZ=75.09* SIN(281.06*PI/180.) -2.
      YZ=75.09* COS(281.06*PI/180.)  -2.
      TN1=0.
```

```
      CALL RGAGE
      CALL BOX(IN,JN)
  194 DO 201 I=2,IN1
  201 READ(2)(RESP(I,J),J=2,JN1)
      READ(2)TDIFF,TDA1,TH1,TM1,ELMAX,ELMIN
      IF(TH1.LT.14.)GO TO 194
      DO 405 I=1,IN
      DO 405 J=1,JN
  405 RESP(I,J)=RESP(I,J)/7.13
      CALL MAPP(RESP,IM,JM)
      CALL FOURAN(RESP,IM,JM,THRESH,COF,PERIM,RFALL,NECH,NHAM, AREA,
     1     4,IBRDS,IECH)
      CALL METRIC(COF,NHAM,4,30,IECH,FMET1,3,SQ1)
      CALL BOX(IN,JN)
      NECH1=IECH
      DO 35 I1=1,NECH1
      NUM(I1)=I1
      IFTN(I1)=I1
      IDENT(I1)=100000*TH1+1000*TM1+10*I1
      XC1(I1)=COF(1,1,I1)
      YC1(I1)=COF(1,3,I1)
      RFAL1(I1)=RFALL(I1)
      DO 34 K=1,3
      XL(K,I1)=XC1(I1)
      YL(K,I1)=YC1(I1)
   34 TL(K,I1)=TH1+TM1/60.
   35 AREA1(I1)=AREA(I1)
      DO 20 IJK=1,100
      DO 202 I=2,IN1
  202 READ(2)(RESP(I,J),J=2,JN1)
      DO 404 I=1,IN
      DO 404 J=1,JN
  404 RESP(I,J)=RESP(I,J)/7.13
      READ(2)TDIFF,TDA1,TH1,TM1,ELMAX,ELMIN
      CALL FOURAN(RESP,IM,JM,THRESH,COF,PERIM,RFALL,NECH,NHAM, AREA,
     1     4,IBRDS,IECH)
      CALL METRIC(COF,NHAM,4,30,IECH,FMET2,3,SQ2)
      NECH2=IECH
```

```
      DO 36 I2=1,NECH2
      XC2(I2)=COF(1,1,I2)
      YC2(I2)=COF(1,3,I2)
      RFAL2(I2)=RFALL(I2)
*  36 AREA2(I2)=AREA(I2)
      CALL MATCH(AREA1,AREA2,XC1,YC1,XC2,YC2,NECH1,NECH2,RFAL1,30
     1       ,FMET1,FMET2,SQ1,SQ2,3)
      CALL MAPP(RESP,IM,JM)
      CALL NUMBER(10.,-4.,1.5,TDA1,0.,-1)
      CALL NUMBER(17.,-4.,1.5,TH1,0.,-1)
      CALL NUMBER(24.,-4.,1.5,TM1,0.,-1)
      DO 120 I=1,NECH2
      A=COF(2,1,I)
      B=COF(2,2,I)
      C=COF(2,3,I)
      D=COF(2,4,I)
      COSX(I)=A*A*FMET2(1,I)+A*C*FMET2(2,I)+C*C*FMET2(3,I)
      SINX(I)=B*B*FMET2(1,I)+B*D*FMET2(2,I)+D*D*FMET2(3,I)
      CROS(I)=2.*A*B*FMET2(1,I)+(B*C+A*D)*FMET2(2,I)+2.*C*D*FMET2(3,I)
  120 CONTINUE
      CALL BOX(IN,JN)
      NECH1=NECH2
      DO 37 I1=1,NECH1
      XC1(I1)=XC2(I1)
      YC1(I1)=YC2(I1)
      RFAL1(I1)=RFAL2(I1)
      DO 39 K=1,3
   39 FMET1(K,I1)=FMET2(K,I1)
   37 AREA1(I1)=AREA2(I1)
   20 CONTINUE
   21 CONTINUE
      CALL EXIT
      END
```

(Find the Contour Starting Point)

```
      SUBROUTINE FOURAN(Z,IN,JN, TRESH ,COF,PERIM,RFALL,NECH,NHEM,AREA,
     1  K4,IBRDS,IECHS)
      LOGICAL DON
      DIMENSION AREA(NECH)
      DIMENSION Z(IN,JN),JCK(4),ICK(4),INC(4),JNC(4),ND(4)
     1 ,COF(NHEM,K4,NECH),PERIM(NECH),RFALL(NECH),IBRDS(NECH)
      COMMON/PLO/IBUF(1024)
      COMMON/ANA/ICNT,IECH,IDON(999),JDON(999),ISDON(999),
     1 ISTS,JSTS,XS(400),YS(400),MCOUNT,ISTORE,
     2 JSTORE,ISIRE,IBORDS,IBORDE,NHAM,IND,JND,THRESH
      DATA JCK/0,0,1,0 /
      DATA ICK/0,1,0,0 /
      DATA JNC/0,1,0,1 /
      DATA INC/1,0,1,0 /
      THRESH=TRESH
      IND=IN
      JND=JN
      NHAM=NHEM
      ND(1)=IN
      ND(2)=JN
      ND(3)=IN
      ND(4)=JN
      DO 1 I=1,IN
      DO 1 J=1,JN
      Z(I,J)=Z(I,J)-THRESH
      IF(ABS(Z(I,J)).GT..1)GO TO 1
      IF(Z(I,J).LT.0.)GO TO 5
      Z(I,J)=.1
      GO TO 1
    5 Z(I,J)=-.1
    1 CONTINUE
      ICNT=1
      IECH=0
      IDON(1)=0
      JDON(1)=0
      ISDON(1)=0
      DO 100 ISI=1,4
      IEND=ND(ISI)
```

```
      DO 50 K=1,IEND
      I=K*INC(ISI)+1+(IN-1)*ICK(ISI)-INC(ISI)
      J=K*JNC(ISI)+1+(JN-1)*JCK(ISI)-JNC(ISI)
      Z(I,J)=-THRESH
   50 CONTINUE
  100 CONTINUE
      IN1=IN-1
      JN1=JN-1
      DO 200 IG=1,IN1
      I=IN-IG
      DO 200 J=2,JN1
      IF(Z(I,J)*Z(I+1,J).GT.0.)GO TO 200
      ISTORE=I
      JSTORE=J
      ISIRE=1
      IBORDS=5
      IF(DON(I,J,1,6))GO TO 200
      CALL FOLL(Z,IN,JN,COF,PERIM,RFALL,NECH,NHEM,K4,IBRDS,AREA)
      IF(IECH.GT.30)GO TO 39
      IF(ICNT.GT.999)WRITE(6,555)
  555 FORMAT(1H0,8HICNT MAX)
      IF(ICNT.GT.999)RETURN
  200 CONTINUE
  203 CONTINUE
      DO 201 I=1,IN
      DO 201 J=1,JN
      Z(I,J)=Z(I,J)+THRESH
  201 CONTINUE
      DO 31 K=1,IECH
      CALL FOUPLO(COF(1,1,K    ),NHAM,4,PERIM(IECH),IBRDS(IECH),K,IN,JN)
   31 CONTINUE
      IECHS=IECH
      RETURN
   39 CONTINUE
      WRITE(6,40)
   40 FORMAT(1H ,20HTOTAL ECHOES TO MANY   )
      GO TO 203
      END
```

(Follow Echo Contour)

```
[FOR,SI   ,.SUB2
      SUBROUTINE FOLL(Z,IN,JN,COF,PERIM,RFALL,NECH,NHEM,K4,IBRDS,AREA)
      LOGICAL DON
      DIMENSION AREA(NECH)
      DIMENSION Z(IN,JN),IC1(4),JC1(4),IC2(4),JC2(4),KX(4),KY(4)
     1,IGT(4),JGT(4),ISGT(4),COF(NHEM,K4,NECH),IBRDS(NECH),RFALL(NECH),
     2 PERIM(NECH)
      COMMON/PLO/IBUF(1024)
      COMMON/ANA/ICNT,IECH,IDON(999),JDON(999),ISDON(999),
     1 ISTS,JSTS,XS(400),YS(400),MCOUNT,ISTORE,
     2 JSTORE,ISIRE,IBORDS,IBORDE,NHAM,IND,JND,THRESH
      DATA ISGT/3,4,1,2/
      DATA IC1 /0,1,0,0/
      DATA JC1 /0,0,1,0/
      DATA IC2 /1,1,1,0/
      DATA JC2 /0,1,1,1 /
      DATA  KX /1,0,1,0/
      DATA  KY /0,1,0,1/
      DATA IGT /0,1,0,-1/
      DATA JGT /-1,0,1,0/
      PI=3.14159
      ISI=ISIRE
      IST=ISTORE
      JST=JSTORE
      ITRY=IST-KX(ISI)
      JTRY=JST-KY(ISI)
      DF=1.E-20
      IECH=IECH+1
      IBRDS(IECH)=IBORDS
      IF(Z(IST,JST).LT.0.)GO TO 6
      ISTART=IST
      JSTART=JST
      GO TO 7
    6 ISTART=IST+KX(ISI)
      JSTART=KY(ISI)+JST
    7 CONTINUE
      M=0
      IF(IBORDS.EQ.2)IST=IN-1
      IF(IBORDS.EQ.2)ISI=2
      IF(IBORDS.EQ.3)JST=JN-1
      IF(IBORDS.EQ.3)ISI=3
      IR1=IST+IC1(ISI)
      JR1=JST+JC1(ISI)
      Z1=Z(IR1,JR1)
      IR2=IST+IC2(ISI)
      JR2=JST+JC2(ISI)
      Z2=Z(IR2,JR2)
      ISTS=IST
      JSTS=JST
```

```
      M=M+1
      XS(M)=IST+IC1(ISI)+Z1/(Z1-Z2)*KX (ISI)
      YS(M)=JST+JC1(ISI)+Z1/(Z1-Z2)*KY(ISI)
      CALL PUT(IST,JST,ISI)
   40 CONTINUE
      DO 50 KZ=1,3
      KS=MOD(KZ+ISI-1,4)+1
      I1=IST+IC1(KS)
      J1=JST+JC1(KS)
      I2=IST+IC2(KS)
      J2=JST+JC2(KS)
      IF(I1.LT.1.OR.I1.GT.IN.OR.J1.LT.1.OR.J1.GT.JN)GO TO 997
      IF(I2.LT.1.OR.I2.GT.IN.OR.J2.LT.1.OR.J2.GT.JN)GO TO 997
      Z1=Z(I1,J1)
      Z2=Z(I2,J2)
      IF(Z1*Z2.GT.0.)GO TO 50
      IF(ABS(Z1-Z2).LE.0.)GO TO 50
      M=M+1
      IF(M.GT.4)GO TO 45
      IF(.NOT.DON(IST,JST,KS,6))GO TO 45
      IECH=IECH-1
      RETURN
   45 CONTINUE
      IF(M.GT.400)WRITE(6,995)
  995 FORMAT(1H0,13HPROBABLE LOOP)
      IF(M.GT.400)GO TO 999
      XS(M)=I1+Z1/(Z1-Z2)*KX( KS )
      YS(M)=J1+Z1/(Z1-Z2)*KY(KS)
      IF(DON(IST,JST,KS,5))GO TO 999
      CALL PUT(IST,JST,KS)
      IF(ICNT.GT.999)GO TO 999
      ISI=ISGT(KS)
      IST=IGT(KS)+IST
      JST=JST+JGT(KS)
      GO TO 40
   50 CONTINUE
      WRITE(6,998)
  998 FORMAT(1H0,10X,16HCANT EXIT SQUARE)
      GO TO 999
  997 WRITE(6,996)I1,I2,J1,J2
  996 FORMAT(1H0,10X,4I5,9HBAD INDEX)
      WRITE(6,990)IBORDS,IST,JST,KS,M
  990 FORMAT(1H ,10X,5I5)
      WRITE(6,991)(XS(MD),MD=1,M)
      WRITE(6,991)(YS(MD),MD=1,M)
  991 FORMAT(1H ,26F5.1)
      IB=MAX0(IST-5,1)
      IE=MIN0(IST+5,IN)
      JB=MAX0(JST-5,1)
      JE=MIN0(JST+5,JN)
```

```
      DO 892 J=JB,JE
      JR=JE+1-J
  892 WRITE(6,895)(Z(I,JR),I=IB,IE)
  895 FORMAT(1H0,/,10F10.5)
  999 CONTINUE
      M=M+1
      XS(M)=XS(1)
      YS(M)=YS(1)
      MCOUNT=M
      IBORDE=KS
      CALL ECHOF(COF,PERIM,NHEM,K4,NECH)
      CALL PUT(IST,JST,KS)
      IF(ICNT.GT.999)RETURN
      I=IECH
      AREA(I)=0.
      DO  160 NH=1,NHAM
  160 AREA(I)=AREA(I)+PI*(COF(NH,1,I)*COF(NH,4,I)-COF(NH,2,I)*
     1   COF(NH,3,I))
      IF(AREA(IECH).GT.0..OR.IBORDS.NE.5)GO TO 161
      IECH=IECH-1
      RETURN
  161 AREA(IECH)=ABS(AREA(IECH))
      IF(AREA(IECH).GT.2.)GO TO 162
      IECH=IECH-1
      RETURN
  162 CONTINUE
      RFALL(I)    =RF(ISTART,JSTART,Z,IN,JN,THRESH)
      RETURN
      END
```

(Store locations already found)

```
      SUBROUTINE PUT(IST,JST,ISI)
      DIMENSION ISP(4),IPU(4),JPU(4)          ,IGT(4),JGT(4)
      COMMON/ANA/ICNT,IECH,IDON(999),JDON(999),ISDON(999),
     1 ISTS,JSTS,XS(400),YS(400),MCOUNT,ISTORE,
     2 JSTORE,ISIRE,IBORDS,IBORDE,NHAM,IND,JND,THRESH
      DATA IGT/1,0,-1,0 /
      DATA JGT / 0,1,0,-1 /
      DATA ISP/1,4,1,4/
      DATA IPU/0,1,0,0/
      DATA JPU/0,0,1,0/
      ICNT=ICNT+1
      IF(ICNT.GT.999)RETURN
      ISDON(ICNT)=ISP(ISI)
      IDON(ICNT)=IST+IPU(ISI)
      JDON(ICNT)=JST+JPU(ISI)
      RETURN
      END
```

(See if a point has already been found)

```
      FUNCTION DON(IST,JST,ISI,KEY)
      LOGICAL DON
      DIMENSION ISP(4),IPU(4),JPU(4)
      COMMON/ANA/ICNT,IECH,IDON(999),JDON(999),ISDON(999),
     1 ISTS,JSTS,XS(400),YS(400),MCOUNT,ISTORE,
     2 JSTORE,ISIRE,IBORDS,IBORDE,NHAM,IND,JND,THRESH
      DATA ISP/1,4,1,4/
      DATA  IPU/0,1,0,0/
      DATA JPU/0,0,1,0/
      DON=.FALSE.
      IF(KEY.EQ.5)GO TO 100
      DO 20 K=1,ICNT
      IF(ISP(ISI).NE.ISDON(K).OR.IST.NE.IDON(K)
     1.OR.JST.NE.JDON(K))GO TO 20
      DON =.TRUE.
      RETURN
   20 CONTINUE
      RETURN
  100 IF(IBORDS.NE.5)GO TO 200
      IF(ISTS.NE.IST+IPU(ISI).OR.JSTS.NE.JST+JPU(ISI)
     1.OR.ISP(ISI).NE.1)GO TO 150
      DON=.TRUE.
  150 RETURN
  200 IF(IST.EQ.IND-1.AND.ISI.EQ.2)DON=.TRUE.
      IF(JST.EQ.JND-1.AND.ISI.EQ.3)DON=.TRUE.
      IF(JST.EQ.1.AND.ISI.EQ.1)DON=.TRUE.
      IF(IST.EQ.1.AND.ISI.EQ.4)DON=.TRUE.
      IF(DON)CALL PUT(IST,JST,ISI)
      RETURN
      END
```

(Calculate FMET, the coefficients describing the ellipse)

```
      SUBROUTINE METRIC(COF,NHAM,K4,K30,IECH,FMET,K3,SQ)
      DIMENSION BAS(3,3),CAS(3,3)
      DIMENSION SQ(K30)
      DIMENSION COF(NHAM,K4,K30),FMET(K3,K30)
      DO 151 I=1,IECH
      A=COF(2,1,I)
      B=COF(2,2,I)
      C=COF(2,3,I)
      D=COF(2,4,I)
      BAS(1,1)=A*A
      BAS(1,2)=A*C
      BAS(1,3)=C*C
      BAS(2,1)=B*B
      BAS(2,2)=B*D
      BAS(2,3)=D*D
      BAS(3,1)=A*B
      BAS(3,2)=(B*C+A*D)/2.
      BAS(3,3)=C*D
      DENOM=BAS(1,1)*(BAS(2,2)*BAS(3,3)-BAS(3,2)*BAS(2,3))
     1      -BAS(1,2)*(BAS(2,1)*BAS(3,3)-BAS(3,1)*BAS(2,3))
     2      +BAS(1,3)*(BAS(2,1)*BAS(3,2)-BAS(2,2)*BAS(3,1))
      SQ(I)=SQRT((A*A+B*B+C*C+D*D) /2.0 )
      DO 151 K=1,3
      DO 143 ID=1,3
      DO 143 J=1,3
  143 CAS(ID,J)=BAS(ID,J)
      CAS(1,K)=SQ(I)
      CAS(2,K)=SQ(I)
      CAS(3,K)=0.
      FNU  =CAS(1,1)*(CAS(2,2)*CAS(3,3)-CAS(3,2)*CAS(2,3))
     1      -CAS(1,2)*(CAS(2,1)*CAS(3,3)-CAS(3,1)*CAS(2,3))
     2      +CAS(1,3)*(CAS(2,1)*CAS(3,2)-CAS(2,2)*CAS(3,1))
  151 FMET(K,I)=FNU/DENOM
      RETURN
      END
```

(Calculate Fourier Coefficients)

```
{FOR.SI` ..SUH/
      SUBROUTINE ECHOF(COF,PERIM,NHEM,K4,NECH)
      DIMENSION COF(NHEM,K4,NECH),PERIM(NECH)
      COMMON/ANA/ICNT,IECH,IDON(999),JDON(999),ISDON(999),
     1 ISTS,JSTS,XS(400),YS(400),MCOUNT,ISTORE,
     2 JSTORE,ISIRE,IBORDS,IBORDE,NHAM,IND,JND,THRESH
      DF=1.E-4
      PI=3.1415926
      DO 1 J=1,NHAM
      DO 1 K=1,4
    1 COF(J,K,IECH)=0.
      SLENG=0.
      DO 6 I=2,MCOUNT
    6 SLENG=SLENG+SQRT((XS(I)-XS(I-1))**2+(YS(I)-YS(I-1))**2)
      PERIM(IECH)=SLENG
      DO 200 NH=2,NHAM
      N=NH-1
      S1=0.
      DO 100 I=2,MCOUNT
      S2=S1+SQRT((XS(I)-XS(I-1))**2+(YS(I)-YS(I-1))**2)
      IF(ABS(S1-S2).LE.DF)GO TO 100
      PX=(XS(I)-XS(I-1))/(S2-S1)
      PY=(YS(I)-YS(I-1))/(S2-S1)
      QX=XS(I)-PX*S2
      QY=YS(I)-PY*S2
      CN=2*N*PI/SLENG
      U1=CN*S1
      U2=CN*S2
      SIN1=SIN(U1)
      SIN2=SIN(U2)
      COS1=COS(U1)
      COS2=COS(U2)
```

```
C                         DO COEFF COS FOR X
      T2=QX/CN*SIN2+PX*(CN**(-2)*COS2+S2/CN*SIN2)
      T1=QX/CN*SIN1+PX*(CN**(-2)*COS1+S1/CN*SIN1)
      COF(NH,1,IECH)=COF(NH,1,IECH)+2./SLENG*(T2-T1)
C                         DO COEFF SIN FOR X
      T2=-QX/CN*COS2+PX*(CN**(-2)*SIN2-S2/CN*COS2)
      T1=-QX/CN*COS1+PX*(CN**(-2)*SIN1-S1/CN*COS1)
      COF(NH,2,IECH)=COF(NH,2,IECH)+2./SLENG*(T2-T1)
C                         DO COEFF COS FOR Y
      T2=QY/CN*SIN2+PY*(CN**(-2)*COS2+S2/CN*SIN2)
      T1=QY/CN*SIN1+PY*(CN**(-2)*COS1+S1/CN*SIN1)
      COF(NH,3,IECH)=COF(NH,3,IECH)+2./SLENG*(T2-T1)
C                          DO COEFF SIN FOR Y COORDS
      T2=-QY/CN*COS2+PY*(CN**(-2)*SIN2-S2/CN*COS2)
      T1=-QY/CN*COS1+PY*(CN**(-2)*SIN1-S1/CN*COS1)
      COF(NH,4,IECH)=COF(NH,4,IECH)+2./SLENG*(T2-T1)
      S1=S2
  100 CONTINUE
  200 CONTINUE
C                         CALCULATE ZERO HARMONUC COEFF
      COF(1,2,IECH)=0.
      COF(1,4,IECH)=0.
      S1=0.
      DO 300 I=2,MCOUNT
      S2=SQRT((XS(I)-XS(I-1))**2+(YS(I)-YS(I-1))**2)  +S1
      IF(ABS(S1-S2).LE.DF)GO TO 300
      PX=(XS(I)-XS(I-1))/(S2-S1)
      PY=(YS(I)-YS(I-1))/(S2-S1)
      QX=XS(I)-PX*S2
      QY=YS(I)-PY*S2
      T2=.5*PX*S2**2+QX*S2
      T1=.5*PX*S1**2+QX*S1
      COF(1,1,IECH)=COF(1,1,IECH)+(T2-T1)/SLENG
      T2=.5*PY*S2**2+QY*S2
      T1=.5*PY*S1**2+QY*S1
      COF(1,3,IECH)=COF(1,3,IECH)+(T2-T1)/SLENG
      S1=S2
  300 CONTINUE
      RETURN
      END
```

(Volume Rainrate Calculation (RF))

```
[FOR,SI  *.SUBIT
      FUNCTION RF(IST,JST,Z,IN,JN,THRESH)
      DIMENSION Z(IN,JN),ICH(8),JCH(8),ICHE(500),JCHE(500)
      DATA ICH /-1,0,1,1,1,0,-1,-1 /
      DATA JCH /-1,-1,-1,0,1,1,1,0 /
      DF=1.E-20 <-- RF=0.
      ICHE(1)=IST
      JCHE(1)=JST
      RF=Z(IST,JST)
      MD=1
      MC=1
      Z(IST,JST)=-Z(IST,JST)
    3 DO 5 I=1,8
      IC=IST+ICH(I)
      JC=JST+JCH(I)
      IF(IC.LT.1.OR.IC.GT.IN.OR.JC.LT.1.OR.JC.GT.JN)GO TO 5
      IF(Z(IC,JC).LT.DF)GO TO 5
      MC=MC+1
      IF(MC.LE.500)GO TO 11
      RF=-1.
      GO TO  8
   11 ICHE(MC)=IC
      JCHE(MC)=JC
      RF=RF+Z(IC,JC)
      Z(IC,JC)=-Z(IC,JC)
    5 CONTINUE
      IF(MD.GE.MC)GO TO 8
      MD=MD+1
      IST=ICHE(MD)
      JST=JCHE(MD)
      GO TO 3
    8 DO 9 M=1,MD
      IC=ICHE(M)
      JC=JCHE(M)
    9 Z(IC,JC)=-Z(IC,JC)
      IF(MD.EQ.500)RETURN
      RF=RF+MD*THRESH
      RETURN
      END
```

(Match Echoes and Update Echo Labels)

```
[FOR,SI   ,.SUB15
      SUBROUTINE MATCH(AREA1,AREA2,XC1,YC1,XC2,YC2,NECH1,
     1   NECH2,RFALL,K30,FMET1,FMET2,SQ1,SQ2,K3)
      DIMENSION FMET1(K3,K30),FMET2(K3,K30),SQ1(K30),SQ2(K30)
      DIMENSION NUMN(30),IETNN(30),IDENTN(30)
      COMMON/MAT/MATCH1(30,6),MATCH2(30,6),NUM(30),IETN(30),IDENT(30),
     1 RTOT(30),MERGN(6 ),MERGI(6 ),NSPL(6)
     2      ,XL(3,30),YL(3,30),TL(3,30)
      COMMON/PLO/IBUF(1024)
      DIMENSION AREA1(NECH1),AREA2(NECH2),XC1(NECH1),YC1(NECH1)
     1      ,XC2(NECH2),YC2(NECH2)  ,RFALL(K30),RBE(5),IRB(5)
      COMMON /BLOK/IBINS,IN ,JN ,DIS,PI,XZ,YZ,TDIFF,TN1,BN1(300),
     1BN2(300),KCART,PHI1,AZM1,TDA1,TH1,TM1,ELMAX,ELMIN ,TE,TDA,TH,TM
      FYX=60.
      TM=TH1+TM1/60.
      PI=3.14159
      DO 57 I=1,30
      DO 57 K=1,6
      MATCH1(I,K)=0
      MATCH2(I,K)=0
   57 CONTINUE
      DO 56  I1=1,NECH1
       IET=IETN(I1)
      PX=(XC1(I1)-XL(3,IET))*TDIFF/(TM-TL(3,IET))
      PY=(YC1(I1)-YL(3,IET))*TDIFF/(TM-TL(3,IET))
      RD=SQRT(PX*PX+PY*PY)
      IF(RD.LT.3.)GO TO  54
      PX=PX*2./RD
      PY=PY*2./RD
   54 CONTINUE
      XC1(I1)=XC1(I1)+PX
      YC1(I1)=YC1(I1)+PY
   56 CONTINUE
      KTMS=0
    1 CONTINUE
      KTMS=KTMS+1
      DO 90 I1=1,NECH1
      QB=1.E20
      RB=1.E20
      IM=0
      IBR=0
      IBQ=0
      DO 5 I2=1,NECH2
      ARM=AMIN1(AREA1(I1),AREA2(I2))
      R=(XC1(I1)-XC2(I2))**2+(YC1(I1)-YC2(I2))**2
      Q=AREA1(I1)/AREA2(I2)
      Q=AMAX1(Q,1./Q)
      IF(R.GT.RB)GO TO 3
```

```
      IBR=I2
      QBR=Q
      RB=R
      QCRITR=1.3+10./ARM
      RCRITR=ARM/PI
    3 CONTINUE
      IF(Q.GT.QB)GO TO 5
      IBQ=I2
      QB=Q
      RBQ=R
      RCRITQ=ARM/PI
      QCRITQ=1.5+30./ARM
    5 CONTINUE
      IM=0
      IF(RBQ.LT.RCRITQ.AND.QB.LT.QCRITQ)IM=IBQ
      IF(RB.LT.RCRITR.AND.QBR.LT.QCRITR)IM=IBR
      MATCH1(I1,1)=IM
      IF(IM.EQ.0)GO TO 90
      MATCH2(IM,1)=I1
   90 CONTINUE
C              LOOK FOR SPLITS
      DO 201 I1=1,NECH1
      RCRIT=(SQ1(I1)+1.)**2
      A=FMET1(1,I1)
      B=FMET1(2,I1)
      C=FMET1(3,I1)
      DO 105 K=1,5
      IRB(K)=0
      MATCH1(I1,K+1)=0
  105 RBE(K)=1.E20
      DO 190 I2=1,NECH2
      IF(MATCH2(I2,1).NE.I1.AND.MATCH2(I2,1).NE.0)GO TO 190
      DX=XC1(I1)-XC2(I2)
      DY=YC1(I1)-YC2(I2)
      R=A*DX*DX+B*DX*DY+C*DY*DY
      IF(R.GT.RBE(5))GO TO 190
      DO 110 K=1,5
      KEND=K
  110 IF(R.LT.RBE(K))GO TO 115
  115 CONTINUE
      K=KEND
      KD=6-KEND
      DO 130 KZ=2,KD
      RBE(7-KZ)=RBE(6-KZ)
  130 IRB(7-KZ)=IRB(6-KZ)
      RBE(K)=R
      IRB(K)=I2
  190 CONTINUE
C          HAVE NOW 5 BEST MATCHES WRT RADIUS, HAVING DISQ. OLD ASSGTS
```

```
      K=1
      IK=IRB(K)
      IF(RBE(1).GT.RCRIT)GO TO 146
      IK=IRB(1)
      AREA=AREA2(IK)
      XC=XC2(IK)
      YC=YC2(IK)
      RB=RBE(1)
      DO 140 K=2,5
      KEND=K
      IK=IRB(K)
      IF(IK.EQ.0)GO TO 142
      IF(RBE( K).GT.RCRIT)GO TO 141
      AREAN=AREA+AREA2(IK)
      XCN=(XC*AREA+XC2(IK)*AREA2(IK))/AREAN
      YCN=(YC*AREA+YC2(IK)*AREA2(IK))/AREAN
      DX=XC1(I1)-XCN
      DY=YC1(I1)-YCN
     RN=A*DX*DX+B*DY*DX+C*DY*DY
     IF(RN.LT.RB)GO TO 132
     IRB(K)=0
     GO TO 140
 132 RB=RN
     AREA=AREAN
     XC=XCN
     YC=YCN
 140 CONTINUE
     GO TO 142
 141 KEND=KEND-1
 142 CONTINUE
     DO 150 KR=1,KEND
 150 MATCH1(I1,KR+1)=IRB(KR)
 146 CONTINUE
 201 CONTINUE
     DO 301 I=1,30
     DUM=AREA1(I)
     AREA1(I)=AREA2(I)
     AREA2(I)=DUM
     DUM=XC1(I)
     XC1(I)=XC2(I)
     XC2(I)=DUM
     DUM=YC1(I)
     YC1(I)=YC2(I)
     YC2(I)=DUM
     DO 302 K=1,3
     DUM=FMET1(K,I)
     FMET1(K,I)=FMET2(K,I)
 302 FMET2(K,I)=DUM
```

```
      DO 301 K=1,6
      MUD=MATCH1(I,K)
      MATCH1(I,K)=MATCH2(I,K)
      MATCH2(I,K)=MUD
  301 CONTINUE
      MUD=NECH1
      NECH1=NECH2
      NECH2=MUD
      IF(KTMS.EQ.1)GO TO 1
      K6=6
      DO 156  I1=1,NECH1
       IET=IETN(I1)
      PX=(XC1(I1)-XL(3,IET))*TDIFF/(TM-TL(3,IET))
      PY=(YC1(I1)-YL(3,IET))*TDIFF/(TM-TL(3,IET))
      RD=SQRT(PX*PX+PY*PY)
      IF(RD.LT.3.)GO TO 154
      PX=PX*2./RD
      PY=PY*2./RD
  154 CONTINUE
      XC1(I1)=XC1(I1)-PX
      YC1(I1)=YC1(I1)-PY
  156 CONTINUE
C
C                     ANALYZE MATCH1  MATCH2
C                     START OF UPDATE
C
      WRITE(6,7)
    7 FORMAT(1H0,16X,66HIDENTIFICATION  CENTROID     AREA   TOT.RT  TOT.
     1RN        STATUS                           )
      DO 8 K=1,30
      NUMN(K)=0
      DO 850 I1=1,NECH1
  850 IF(MATCH1(I1,2).EQ.0)MATCH1(I1,2)=MATCH1(I1,1)
      DO 851 I2=1,NECH2
  851 IF(MATCH2(I2,2).EQ.0)MATCH2(I2,2)=MATCH2(I2,1)
      IDENTN(K)=0
      IETNN(K)= 0
    8 CONTINUE
      DO 300 IET=1,30
      I1=NUM(IET)
      IF(I1.LE.0)GO TO 300
      RTOT(IET)=RTOT(IET)+TDIFF*RFALL(I1)
C                     CHECK 1-1 MATCHES
      IF(MATCH1(I1,2).EQ.0)GO TO 100
      DO 10 K=3,6
   10 IF(MATCH1(I1,K).NE.0)GO TO 100
      I2M=MATCH1(I1,2)
      IF(MATCH2(I2M,2).NE.I1)GO TO 100
      DO 11 K=3,6
   11 IF(MATCH2(I2M,K).NE.0)GO TO 100
```

```
      NUMN(IET)=I2M
      IETNN(I2M)=IET
      IDENTN(IET)=IDENT(IET)
      DX=XC2(I2M)-XL(3,IET)
      DY=YC2(I2M)-YL(3,IET)
      SPD=SQRT(DX*DX+DY*DY)/(TM-TL(3,IET)   )
      ANG=ATAN2(DY,DX)*180./PI
      ANG=90.-ANG
      IF(ANG.LT.0.)ANG=ANG+360.
      WRITE(6,109)IDENT(IET),IET,XC1(I1),YC1(I1),AREA1(I1),RFALL(I1),
     1      RTOT(IET)  ,ANG,SPD
  109 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,F8.1,F9.1,5X,
     1      26HTRACKING OLD ECHO HEADING ,F6.1,11H DEGREES AT  ,F4.1,6H K
     1NOTS       )
      CALL NUMBER(-20.,FYX,1.,FLOAT(IET)+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HTR,0.,2)
      FYX=FYX-2.
      DO 111 K=1,2
      KN=4-K
      XL(KN,IET)=XL(KN-1,IET)
      YL(KN,IET)=YL(KN-1,IET)
  111 TL(KN-1,IET)=TL(KN-1,IET)
      TL(1,IET)=TM
      XL(1,IET)=XC2(I2M)
      YL(1,IET)=YC2(I2M)
      GO TO 300
  100 CONTINUE
C                    CHECK FOR TOTALLY LOST ECHOES
      IF(MATCH1(I1,2).NE.0)GO TO 200
      DO 120 I2=1,NECH2
      DO 120 K=2,6
  120 IF(MATCH2(I2,K).EQ.I1)GO TO 200
      WRITE(6,129)IDENT(IET),IET,XC1(I1),YC1(I1),AREA1(I1),RFALL(I1),
     1      RTOT(IET)
      CALL NUMBER(-20.,FYX,1.,FLOAT(IET)+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HLO,0.,2)
      FYX=FYX-2.
  129 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,F8.1,F9.1,5X,
     1      9HECHO LOST )
      RTOT(IET)=0.
      NUM(IET)=0
       DO 131 K=1,3
      XL(K,IET)=0.
      YL(K,IET)=0.
  131 TL(K,IET)=0.
      GO TO 300
C                    CHECK FOR MERGERS
  200 CONTINUE
      DO 210 K=1,6
      MERGN(K)=0
  210 MERGI(K)=0
```

```
      DO 220 I2=1,NECH2
      DO 220 K=2,6
      IF(MATCH2(I2,K).NE.I1)GO TO 220
      I2S=I2
      KS=K
      GO TO 221
  220 CONTINUE
      GO TO 300
  221 CONTINUE
      DO 223 K=2,6
      IF(K.EQ.KS)GO TO 223
      IF(MATCH2(I2S,K).NE.0)GO TO 224
  223 CONTINUE
      GO TO 300
  224 CONTINUE
      DO 225 K=2,6
      M2=MATCH2(I2S,K)
      IF(M2.EQ.0)GO TO 225
      KS=K
      MERGN(K)=IETN(M2)
      MN=MERGN(K)
      MERGI(K)=IDENT(MN)
      NUM(MN)=0
  225 CONTINUE
      IDENTN(IET)=100000*TH1+1000*TM1+10*I2S+2
      NUMN(IET)=I2S
      IETNN(I2S)=IET
      DO 229 K=2,6
      MN=MERGN(K)
      IF(MN.EQ.0)GO TO 229
      M3=MATCH2(I2S,K)
      RTO=RTOT(MN)+RFALL(M3)*TDIFF
      RTOT(MN)=0.
      DO 230 L=1,3
      XL(L,MN)=0.
      YL(L,MN)=0.
  230 TL(L,MN)=0.
      IP=MERGN(K)
      CALL NUMBER(-20.,FYX,1.,FLOAT(IP )+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HME,0.,2)
      FYX=FYX-2.
      WRITE(6,226)MERGI(K),MERGN(K),XC1(M3),YC1(M3),AREA1(M3),RFALL(M3),
     1      RTOT(MN)
  226 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,F8.1,F9.1,5X,
     1      12HLOST MERGED )
  229 CONTINUE
      WRITE(6,227)IDENTN(IET),IET,XC2(I2S),YC2(I2S),AREA2(I2S)
  227 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,22X,
     1      20HNEW RESULT OF MERGER  )
      CALL NUMBER(-20.,FYX,1.,FLOAT(IET)+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HRM,0.,2)
```

```
      FYX=FYX-2.
      RTOT(IET)=0.
      DO 240 L=1,3
      XL(L,IET)=XC2(I2S)
      YL(L, IET)=YC2(I2S)
  240 TL(L,IET)=TM
  300 CONTINUE
      DO 500 IET=1,30
      I1=NUM(IET)
      IF(I1.LE.0)GO TO 500
      IF(NUMN(IET).NE.0)GO TO 500
C                          CHECK FOR SPLITS
      DO 305 K=2,6
      NSPL(K)=0
  305 CONTINUE
      IF(MATCH1(I1,2).EQ.0)GO TO 400
      DO 310 K=3,6
  310 IF(MATCH1(I1,K).NE.0)GO TO 311
      GO TO 400
  311 CONTINUE
      IB=1
      DO 330 K=2,6
      IF(MATCH1(I1,K).EQ.0)GO TO 330
      DO 315 I=IB,30
      IF(NUMN(I).NE.0)GO TO 315
      IB=I+1
      M1=MATCH1(I1,K)
      NUMN(I)=M1
      IDENTN(I)=100000*TH1+1000*TM1+10*M1+1
      IETNN(M1)=I
      NSPL(K)=I
      GO TO 330
  315 CONTINUE
  330 CONTINUE
      DO 331 K=1,3
      XL(K,IET)=0.
      YL(K,IET)=0.
  331 TL(K,IET)=0.
      CALL NUMBER(-20.,FYX,1.,FLOAT(IET)+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HSP,0.,2)
      FYX=FYX-2.
       WRITE(6,348)IDENT(IET),IET,XC1(I1),YC1(I1),AREA1(I1),RFALL(I1),
     1     RTOT(IET)
  348 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,F8.1,F9.1,5X,
     1   10HLOST,SPLIT  )
```

```
      DO 349 K=2,6
      IF(NSPL(K).EQ.0)GO TO 349
      I=NSPL(K)
      I2=NUMN(I)
      DO 340 L=1,3
      XL(L,I)=XC2(I2)
      YL(L,I)=YC2(I2)
  340 TL(L,I)=TM
      CALL NUMBER(-20.,FYX,1.,FLOAT(I   )+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HRS,0.,2)
      FYX=FYX-2.
      WRITE(6,347)IDENTN(I),I,XC2(I2),YC2(I2),AREA2(I2)
  347 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,22X,
     1       20HNEW RESULT OF SPLIT  )
      RTOT(I)=0.
  349 CONTINUE
      GO TO 500
  400 CONTINUE
      CALL NUMBER(-20.,FYX,1.,FLOAT(IET)+.1,0.,-1)
      CALL SYMBOL(-18.,FYX,1.,2HLO,0.,2)
      FYX=FYX-2.
      WRITE(6,129)IDENT(IET),IET,XC1(I1),YC1(I1),AREA1(I1),RFALL(I1),
     1       RTOT(IET)
  500 CONTINUE
C                       LOOK FOR NEW ECHOES
      IB=1
      DO 700 I2=1,NECH2
      IF(IETNN(I2).GT.0)GO TO 700
      DO 650 I=IB,30
      IF(NUMN(I).GT.0)GO TO 650
      IB=I+1
      NUMN(I)=I2
      IETNN(I2)=I
      IDENTN(I)=100000*TH1+1000*TM1+10*I2
      RTOT(I)=0
      CALL NUMBER(-20.,FYX,1.,FLOAT(I   )+.1,0.,-1)
```

```
      CALL SYMBOL(-18.,FYX,1.,2HNG,0.,2)
      FYX=FYX-2.
      WRITE(6,662)IDENTN(I),I,XC2(I2),YC2(I2),AREA2(I2)
  662 FORMAT(1H ,16X,I9,I4,3H  (,F4.1,1H,,F4.1,1H),F7.1,22X,
     1      8HNEW ECHO )
      DO 670 K=1,3
      XL(K,I)=XC2(I2)
      YL(K,I)=YC2(I2)
  670 TL(K,I)=TM
      GO TO 700
  650 CONTINUE
  700 CONTINUE
      DO 750 I=1,30
      IETN(I)=IETNN(I)
      IDENT(I)=IDENTN(I)
      NUM(I)=NUMN(I)
  750 CONTINUE
      DO 800 I2=1,NECH2
      IT=IETN(I2)
      FN=FLOAT(IT)+.1
      CALL NUMBER(XC2(I2),YC2(I2)    ,1.,FN,0.,-1)
  800 CONTINUE
      RETURN
      END
```

(Draw Box around display)

```
[FOR,SI  ,.SUBB
      SUBROUTINE BOX(IN,JN)
      COMMON/PLO/IBUF(1024)
      CALL PLOT(0.,0.,-3)
      FJN=FLOAT(JN)-1.
      FIN=FLOAT(IN)-1.
      CALL PLOT(2.,2., 3)
      CALL PLOT(2.,FJN,2)
      CALL PLOT(FIN,FJN,2)
      CALL PLOT(FIN,2.,2)
      CALL PLOT(2.,2.,30.)
      CALL PLOT(FIN+30.,0.,-3)
      RETURN
      END
```

(Echo Pen Tracing)

```
[FOR,SI   ,.SUB10
      SUBROUTINE FOUPLO(COEF,NHAM,K4,PERI,IBORDS,IECH,IND,JND)
      DIMENSION COEF(NHAM,K4)
      COMMON/PLO/IBUF(1024)
      FIND=FLOAT(IND)-1.
      FJND=FLOAT(JND)-1.
      JDIS=5
      PI=3.14159236
      IEND=JDIS*PERI
      IE=IEND
      IPEN=3
      IBEG=1
    2 DO 4 I=IBEG,IEND
      Y=COEF(1,3)
      X=COEF(1,1)
      DO 3 N=2,NHAM
      U=2.*(N-1)*PI*I/FLOAT(IE)
      COSU=COS(U)
      SINU=SIN(U)
      X=X+COEF(N,1)*COSU+COEF(N,2)*SINU
      Y=Y+COEF(N,3)*COSU+COEF(N, 4)*SINU
    3 CONTINUE
      IF(X.LT.2..OR.X.GT.FIND.OR.Y.LT.2..OR.Y.GT.FJND)IPEN=3
      CALL PLOT(X,Y,IPEN)
      IPEN=2
    4 CONTINUE
      RETURN
      END
```

(Draw Raingage Network)

```
      SUBROUTINE RGAGE
      COMMON/PLO/IBUF(1024)
      DIMENSION FIX(8),FJX(8),SX( 9),SY( 9),CX(8),CY(8),PX(6),PY(6)
      DIMENSION QX(16),QY(16)
      DATA/FIX/12.6,11.3,12.1,12.1,13.6,15.1,13.8,13.9/
      DATA/FJX/27.5,26.6,26.3,25.0,26.0,27.0,27.0,28.3/
      DATA/ SX/46.6,45.2,44.2,45.2,44.5,42.3,43.2,42.7,41.5/
      DATA/ SY/41.0,39.7,41.0,41.4,42.8,42.7,41.9,41.1,39.7/
      DATA/ CX/36.4,38.2,40.1,40.0,38.8,38.2,36.2,36.7/
      DATA/ CY/20.2,20.3,20.0,18.5,19.6,18.8,19.3,18.3/
      DATA/ PX/55.5,57.1,54.1,55.9,53.5,54.8/
      DATA/ PY/52.2,51.2,50.8,49.6,49.5,47.8/
      DATA/QX/62.9,61.2,58.7,58.7,53.5,53.5,47.5,47.5
     1        ,46.5,46.5,43.5,43.5,44.5,44.5,57.5,62.9/
      DATA/ QY/30.3,27.4,27.4,28.5,28.5,28.1,28.1,29.1
     1         ,29.1,35.0,35.0,37.5,37.5,38.2,38.2,30.3/
      IPEN=3
      SF=2.567
      DO 1 I=1,8
      X=FIX(I)    +1
      Y=FJX(I)+1.
    1 CALL SYMBOL(X,Y,.5,1HI,0.,1)
      DO 2 I=1,9
      X=SX(I) +1.
      Y=SY(I)   +1.
    2 CALL SYMBOL(X,Y,.5,1HS,0.,1)
      DO 3 I=1,8
      X=CX(I) +1.
      Y=CY(I) +1.
    3 CALL SYMBOL(X,Y,.5,1HC,0.,1)
      DO 4 I=1,6
      X=PX(I) +1.
      Y=PY(I) +1.
    4 CALL SYMBOL(X,Y,.5,1HP,0.,1)
      DO 5 I=1,16
      X=QX(I) +1.
      Y=QY(I) +1.
      CALL PLOT(X,Y,IPEN)
    5 IPEN=2
      CALL PLOT(48.84,29.08        )
      RETURN
      END
```


☆GPO 1974 — 784-576/1265 REGION NO. 8