DRA

(NASA-CR-124004) CORRELATION SIGNATURES OF WET SOILS AND SNOWS Final Report, 22 Mar. 1971 - 22 Sep. 1972 (IIT Research Inst.) ~~180~~ 182 p HC $11.00 CSCL 08M G3/13

N74-11214

Unclas
22587

IITRI

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

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

1. ORIGINATING ACTIVITY (Corporate author) IIT Research Institute 10 West 35th Street Chicago, Illinois 60616	2a. REPORT SECURITY CLASSIFICATION 2b. GROUP

3. REPORT TITLE
CORRELATION SIGNATURES OF WET SOILS AND SNOWS

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

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

6. REPORT DATE October 1972	7a. TOTAL NO. OF PAGES 173	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO. Contract NAS8-26797 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) J6243-6 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY NASA George C. Marshall Space Flight Ctr Marshall Space Flight Ctr, Alabama

13. ABSTRACT

SUMMARY

During the past several months and since the inception of the Flight Data Statistics Office, a growing trend to analyze and process Earth Resources Flight Data has prevailed. In view of the growing interest at Marshall Space Flight Center to organize and establish capabilities to perform research studies in this area, endeavors in the areas of interpretation, analysis, and development of algorithms have provided the necessary computational programming tools for data processing and data handling and analysis. Algorithms that have been developed thus far, are adequate and have been proven successful for several preliminary and fundamental applications such as software interfacing capabilities, probability distributions, grey level print plotting, contour plotting, isometric data displays, joint probability distributions, boundary mapping, channel registration and ground scene classification. This report is written in two sections. Section I consists of the algorithms that have been developed individually under the existing contract and section II is a description of an Earth Resources Flight Data Processor, (ERFDP), which handles and processes earth resources data under a users control.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

UNCLASSIFIED
Security Classification

Engineering Mechanics Division
IIT Research Institute
10 West 35th Street
Chicago, Illinois 60616

CORRELATION SIGNATURES OF WET SOILS AND SNOWS

IITRI Final Report J6243-6
Contract NAS8-26797
Period 22 March 1971 to 22 September 1972

M. R. Phillips

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

CORRELATION SIGNATURES OF WET SOILS AND SNOWS

M. R. Phillips

Engineering Mechanics Division
IIT Research Institute
10 West 35th Street
Chicago, Illinois 60616

Final Report
Contract NAS8-26797
Period 22 March 1971 to 22 September 1972
October 1972

Prepared for
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

FOREWORD

This final technical report, entitled "Correlation Signatures of Wet Soils and Snows", presents the results of 18 months programming and algorithm development under the auspices of the National Aeronautics and Space Administration (NASA), with Mr. Robert Jayroe acting COR. This work was performed under contract NAS8-26797 (IITRI Project J6243).

CONTENTS

CONTENTS (concl)

ILLUSTRATIONS

ILLUSTRATIONS (concl)

CORRELATION SIGNATURES OF WET SOILS AND SNOWS

M. R. Phillips
IIT Research Institute

SUMMARY

During the past several months and since the inception of the Flight Data Statistics Office, a growing trend to analyze and process Earth Resources Flight Data has prevailed. In view of the growing interest at Marshall Space Flight Center to organize and establish capabilities to perform research studies in this area, endeavors in the areas of interpretation, analysis, and development of algorithms have provided the necessary computational programming tools for data processing and data handling and analysis. Algorithms that have been developed thus far, are adequate and have been proven successful for several preliminary and fundamental applications such as software interfacing capabilities, probability distributions, grey level print plotting, contour plotting, isometric data displays, joint probability distributions, boundary mapping, channel registration and ground scene classification. This report is written in two sections. Section I consists of the algorithms that have been developed individually under the existing contract and section II is a description of an Earth Resources Flight Data Processor, (ERFDP), which handles and processes earth resources data under a users control.

SECTION I INTRODUCTION

In this section a description of each algorithm developed under the existing contract will be presented. The algorithms were developed as building blocks toward an automatic data processor for processing earth resources flight data. Each algorithm has been demonstrated to be compatible in an overall automatic processing environment. These algorithms were developed using an IBM 7094 computer with 32K available core storage.

1. PROBABILITY DISTRIBUTION

A probability density function was programmed, and the earth resources data processor which is available upon request as an option selection is included. This option gives a probability density function for any number of channels, not to exceed 12, of the multispectral data. This option is used when calculating the slicing intervals of the probability distribution to provide grey level mapping of any ground scene image. An estimate of the probability density function is obtained digitally by dividing the range of the data into a desired number of class intervals d_i. The probability of occurrence of a value X_i in the class interval is then given by

$$P_i(X) = N_i(X)/N,$$

where $N_i(X)$ is the number of values that occur within the range of $d_i < X \leq d_{i+1}$ and

$$N = \sum_{i=0}^{k-1} N_i$$

is the total size of the population where K = number of class intervals.

A. Computer Program

The computer program calculates the probability density of any channel of a multispectral scanner consisting of 12 channels of data. The distribution table is then printed out for all channels requested, over all the data samples (see table I). The program also passes to the grey level module the probability distribution of each channel for grey level mapping.

TABLE I

PROBABILITY DISTRIBUTION

AMPLITUDE	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8	CH 9	CH 10	CH 11	CH 12
-1.5	0	0	0	0	0	0	0	0	0	0	0	0
-1.4	0	0	2	0	2	0	0	0	0	0	1	0
-1.3	0	0	3	0	0	0	0	0	0	0	0	0
-1.2	1	0	9	0	5	0	0	1	0	0	1	0
-1.1	0	0	21	0	7	0	0	0	1	1	8	0
-1.0	5	0	29	0	15	1	0	1	0	0	22	0
- .9	4	0	35	0	17	1	0	3	2	4	31	0
- .8	8	1	39	0	34	2	3	9	7	8	47	0
- .7	12	0	48	0	39	8	8	18	20	16	60	0
- .6	24	7	55	1	47	12	12	22	25	30	55	0
- .5	21	11	66	0	51	22	22	38	40	40	40	0
- .4	38	5	58	2	55	31	41	47	48	52	27	0
- .3	45	12	49	12	49	49	70	50	55	60	11	0
- .2	57	22	42	18	44	66	82	54	62	72	6	5
- .1	59	37	33	20	39	72	96	47	51	55	5	1
0	62	44	25	35	33	62	79	41	45	52	0	8
.1	58	48	19	30	25	40	62	35	39	31	1	16
.2	51	49	12	41	18	22	38	25	25	20	0	27
.3	47	56	6	54	7	12	12	17	16	15	0	32
.4	35	52	1	61	8	8	10	12	8	7	0	39
.5	25	40	0	55	2	2	7	8	5	3	0	47
.6	22	32	1	45	1	0	2	0	1	5	0	52
.7	8	24	0	40	0	1	1	2	0	1	0	65
.8	5	16	0	35	1	0	0	0	0	0	0	72
.9	7	8	0	28	0	0	0	0	0	0	0	58
1.0	3	7	0	20	0	0	0	0	0	0	0	46
1.1	1	1	0	12	0	0	0	0	0	0	0	39
1.2	0	0	0	7	0	0	0	0	0	0	0	22
1.3	1	1	0	0	0	0	0	0	0	0	0	7
1.4	0	0	0	1	0	0	0	0	0	0	0	1
1.5	0	0	0	0	0	0	0	0	0	0	0	0

NUMBER OF OCCURRENCES

B. Data Problem Parameters

NCH	Number of channels or spectral bands on input tape
NSPS	Number of samples per record or resolution elements across one scan
NSCANS	Number of scans to process
NSTART	Starting resolution element
NSTOP	Stopping resolution element
NBTLG	Bit length of input data word
MODE	Signifying FORTRAN or non-FORTRAN input tape
ITYPE	Type of input data; Floating point or fixed point
MSFC	MSFC scanner format option
LTN	Logical unit to load input tape
NSKIP	Number of initial data records to skip before processing
NCRE	Data incrementation
XMAX	Maximum value in data set
XMIN	Minimum value in data set
NOCHS	Number of channels to calculate probability density function
NWHICH	Channel selection for probability distribution calculation

2. GREY LEVEL MAPPING

In order to preview earth resources data and obtain quick-look information, a grey level mapping program was written to include in the ERFDP. This gives a pictorial display of the ground scene image quantized to 10 different levels. Characters are selected to represent varying shades of grey. The levels are calculated by slicing the probability distribution table of selected channels into 10 separate cells (see Figure 1). The number of occurrences in each cell is equally distributed over the probability distribution. This is done by calculating equal areas for each cell using the trapezoidal rule method. Each of the 10 areas would represent one cell of the probability distribution. Each resolution element is compared with all the cells, and the cell in which it falls is assigned the respective alphanumeric character.

Ten Equal Area Intervals

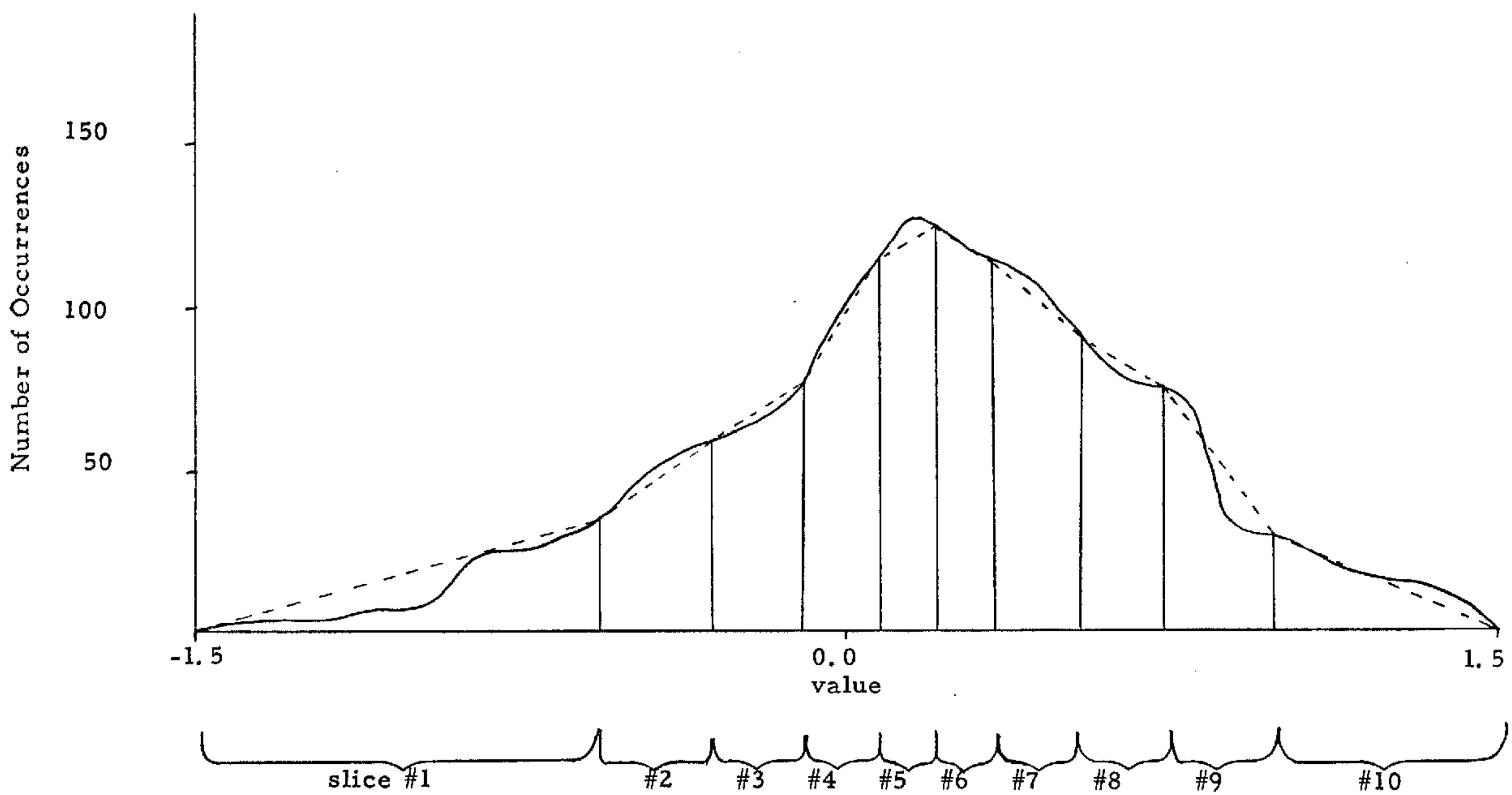


Figure 1 Quantized Probability Distribution Curve

A. Computer Program

The computer program was written as a module to be included in the ERFDP. The logic provides communication with the probability distribution program, since information calculated in that module is used in grey level mapping. The data are input to the program module and each resolution element input to the module is compared with each cell unitl it falls within the quantized level of that cell. Each resolution element, x_i, is examined such that

$$\text{SLICE } (j-1) < x_i \leq \text{SLICE } (j)$$
$$\text{where } j = 1,2,3 \ldots, 11.$$

After the data point, x_i has been quantized by this method, it is replaced by its representative alphanumeric character and plotted or printed. The plotting is output to the Stromberg-Carlson 4020 recorder for display.

If automatic quantization is not desired, the program provides an option where the user can input quantized levels as a table if other methods are desired (reference ERFDP users manual, section II of this report).

B. Data Problem Parameters

NCH	Number of channels on input tape
NSPS	Number of samples per record or scan
NSCANS	Number of scans to process
NSKIP	Number of scans to skip before processing
NSTART	Starting sample number in the scan
NSTOP	Stopping sample number in the scan
ITERM	Number of passes to make processing 120 samples each pass
N	Number of quantized levels plus one
ICHAN	Selected channel used in grey level mapping
IPRT	Option to print grey level

IPLT	Option to plot grey level
INCX	Increment in x direction for each sample (rasters)
INCY	Increment in y direction for each sample (rasters)
NSTX	Starting x coordinates or plot frame
NSTY	Starting y coordinates on plot frame
NBTLG	Bit length of input data words
MODE	Signifies FORTRAN or non-FORTRAN
ITYPE	Type of input data; Floating point or fixed point
MSFC	MSFC scanner format option
NCRE	Data increment
LTN	Logical tape unit to load input data
IOPT	Option to select automatic quantization or input quantized table

3. CONTOUR PLOTTING

In the analysis and interpretation of earth resources data, a useful tool is means of locating and outlining borders of ground scene images and also plot the snow pack temperature profiles recorded by an electronic scanner. A computer program was written to generate line plots, which is a line connecting points of boundaries, specified altitudes, temperatures, etc., providing a graphical display of contour levels or borders of ground scene images.

The data array to be plotted consists of the input data array only. These data points are ordered in an x y coordinate system. Four adjacent data points and their coordinates are examined to determine if a specified value intersects any of these four points. If an intersection occurs, these coordinates are converted to plotting raster counts, and two points are flagged. One point being coordinates of the entry point to the four adjacent data points and the other being the coordinates of the exit point of the four adjacent data points. If the intersection of a specified level continues to the next four successive adjacent data points, then the coordinates of the exit point become the coordinates of the entry point of the next four adjacent data points. A search continues for the coordinates of the exit point and this procedure continues until the data set is exhausted. Lines are drawn connecting the points, reflecting continuity of the specified level. If the entry and exit point of successive four adjacent data points exist, then a line connecting these points is continuous. Otherwise, there exists a discontinuity and the points will not be connected. Continuous lines of multiple levels can be drawn reflecting contours of altitude, temperatures, and homogeneous area boundaries.

A. Computer Program

The computer program was written as a module to be included in the ERFDP and was designed to contour data sets of infinite lengths. However, only 2500 data points reside in the computer at one time because of the physical storage limits. Since only one block of data is processed at one time, the program automatically reloads blocks of data and abuts each block to provide a continuous plot of contours.

In Figure 2 contours of boundaries were plotted of the first 120 scans of field C1 from the Purdue data set. The data were blocked such that contours of 1776 resolution elements were plotted which covered eight scans. Each resolution element covered eight rasters on the plot frame, therefore each block abutted together occupies 64 rasters. After the abutment process, the boundary contours appear continuous. The boundaries that were contoured are shown in Figure 3. The boundary contours are elongated slightly due to scaling on the plot frame.

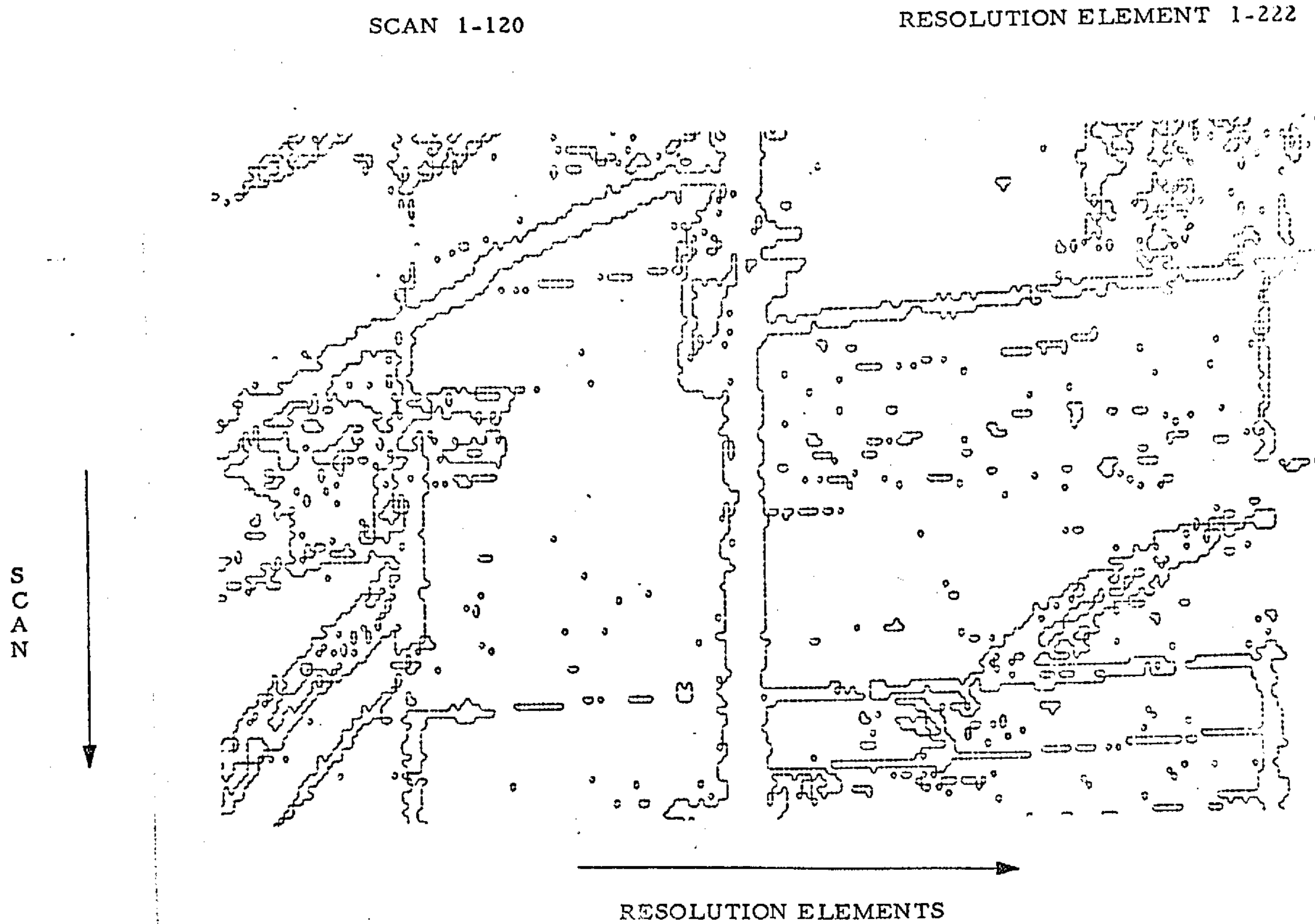


Figure 2 Boundary Contours (Purdue C1)

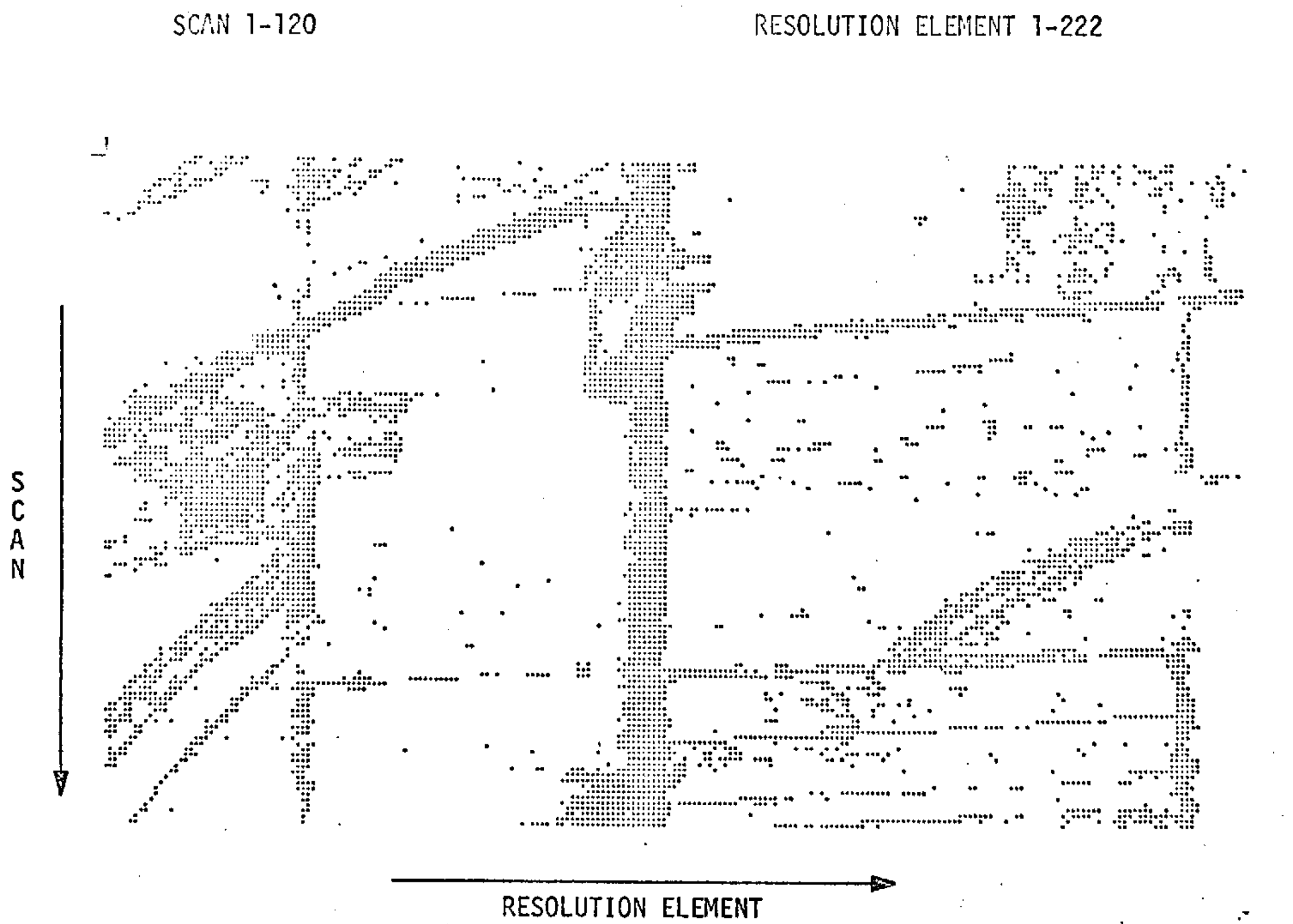


Figure 3 Boundary Determination Channels 4 of 12 (Purdue C1)

B. Data Problem Parameters

NCH	Number of channels on input tape
NSPS	Number of samples in a scan
IRW	Logical tape units to load input data
NCHAN	Channel number
NSNCRE	Number of scans to increment
NPCRE	Number of samples to increment
NPTSL	Lower starting point
NPTSU	Upper stopping point
NOTLG	Bit length of input data word
ITYPE	Type data on input tape; Floating point or fixed point
MSFC	MSFC scanner format option
NSKIP	Number of records to skip
MAXSCN	Total number of scans to process
NSECT	Number of sections to process
MSZX	Data block size in x (samples)
MSZY	Data block size in y (scans)
BLK	Plot frame block size in rasters
FHINC	Contour level increment
ZMIN	Minimum contour level
ZMAX	Maximum contour level
LAB	Label interval

4. ISOMETRIC DISPLAYS

On occasion it has become necessary to preview the raw data and to determine the physical appearance to better understand the behavior of data collected from various ground scenes and from different types of sensors. For this purpose an isometric program was written to produce a two-dimensional projection of a three-dimensional graph of the raw data amplitude versus the x and y ground scene coordinates.

A. Computer Program

Program Isometric will handle any FORTRAN IV formatted tape with up to 12 channels of data. Only one channel is plotted per computer pass and the channel selection is based upon an input parameter.

The program was developed to produce two-dimensional data projections of ground cover scenes where data are gathered by sensors aboard an aircraft. The data collected across the flight line represent resolution elements, and the data collected along the flight line represent scans. Any NXM array can be displayed where N is the position in an array and M is the value to be displayed. The program does not require large amounts of core storage since it performs a continuous operation on the NXM array which is refilled before each operation.

The Stromberg-Carlson 4020 frame reference coordinates are altered after each operation for proper scaling based on the input parameters provided by the user. These parameters also control the density of the points plotted and the angle of rotation desired for display. Multiple plot frames are generated as necessary to display all the input data. To provide frame abutment for a continuous plot, the last row of each frame and the last sample of each row is stored and repeated on the next frame.

B. Data Problem Parameters

NCH	Total number of channels per resolution element on input tape
NSPS	Total number of resolution elements per channel across the ground scene
IRW	Logical tape number for input data
NCHAN	Number of the channel that isometrics are to be displayed
NSNCRE	Number of scans to increment along the flight line; This provides an option to use every scan line, every other scan line, etc.

NPCRE	Number of resolution elements to increment going across the flight line; If every resolution element is not desired, resolution elements can be skipped.
NPTSL	Starting resolution element number for the isometric
NPTSU	Stopping resolution element number for the isometric
MAXSON	Total number of scans along the flight to display
YMIN	The minimum value of the input data used for calculating scale factors for plotting the y axis
YMAX	The maximum value of the input data used also for calculating scale factors for plotting the y axis
NBLSZX	Stromberg-Carlson 4020 reference frame coordinate increment in the x direction
NBLSZY	Stromberg-Carlson 4020 reference frame coordinate increment in the y direction in raster counts (normal range 6 to 12) NBLSZX and NBLSZY are used to determine the degree of rotation of the isometric.
NSECT	Number of passes through the data necessary to display the full-scan width; If NPTSL and NPTSU only cover a portion of the data then NSECT can be adjusted to cover all the data. Example: If NPTSL = 1, NPTSU = 128 and NSPS = 256, then NSECT = 2 will cover resolution element 1 through 128 and resolution element 129 through 256.
NSMOV	Number of points in a moving mean span used in smoothing the input data; Set to zero if smoothing is not desired.
NDIREC	Direction of the rotation of the isometric; 1 = counterclockwise and -1 = clockwise rotation.

A portion of flight line C1 (Purdue data set) was used in the following examples. Every resolution element and every scan was used requiring two passes through the data. NSECT was set to 2, but only one section is shown here. NBLSZX and NBLSZY were set to 8 and 12 respectively and both counterclockwise and clockwise rotations were used.

Figure 4 shows scan lines 86-129 and resolution elements 1-111 of the total 222 elements of channel 8. NDIREC was set to -1 to rotate clockwise. Figure 5 shows channel 8 with only NDIREC changed to +1 to rotate counterclockwise. Figures 6 and 7 reflect identical isometrics using channel 2 of the same data.

5. DYNAMIC JOINT PROBABILITY DISTRIBUTION

Extensive data analysis of data on hand has revealed inadequacies in the current joint probability distribution program. The program was restrictive in that a fixed amount of square storage array allocation was required. This meant that, the greater the spread of the clusters of the joint probabilities or individual data pairs, the larger the square storage array required. This also required a screening of all the data to determine the ranges of data pairs. After determining the data ranges, the minimum square storage array required to display the joint probabilities became NXM, where N is the range between the minimum and maximum value of the X data in the joint pairs, and M is the range between the minimum and maximum value of the Y data in the joint pairs. As this clearly points out, this does not adapt itself favorably to data that is ill behaved and possessing widely spread data clusters or widely spread individual data pairs. Due to the physical limits of the computer storage, in numerous cases, a considerably amount of data was lost. This depended largely on the characteristics of the data and since the primary function of the program was to display the dependent characteristics of two data channels, this required rerunning the program and making adjustments to the limits. Therefore, the program became inadequate.

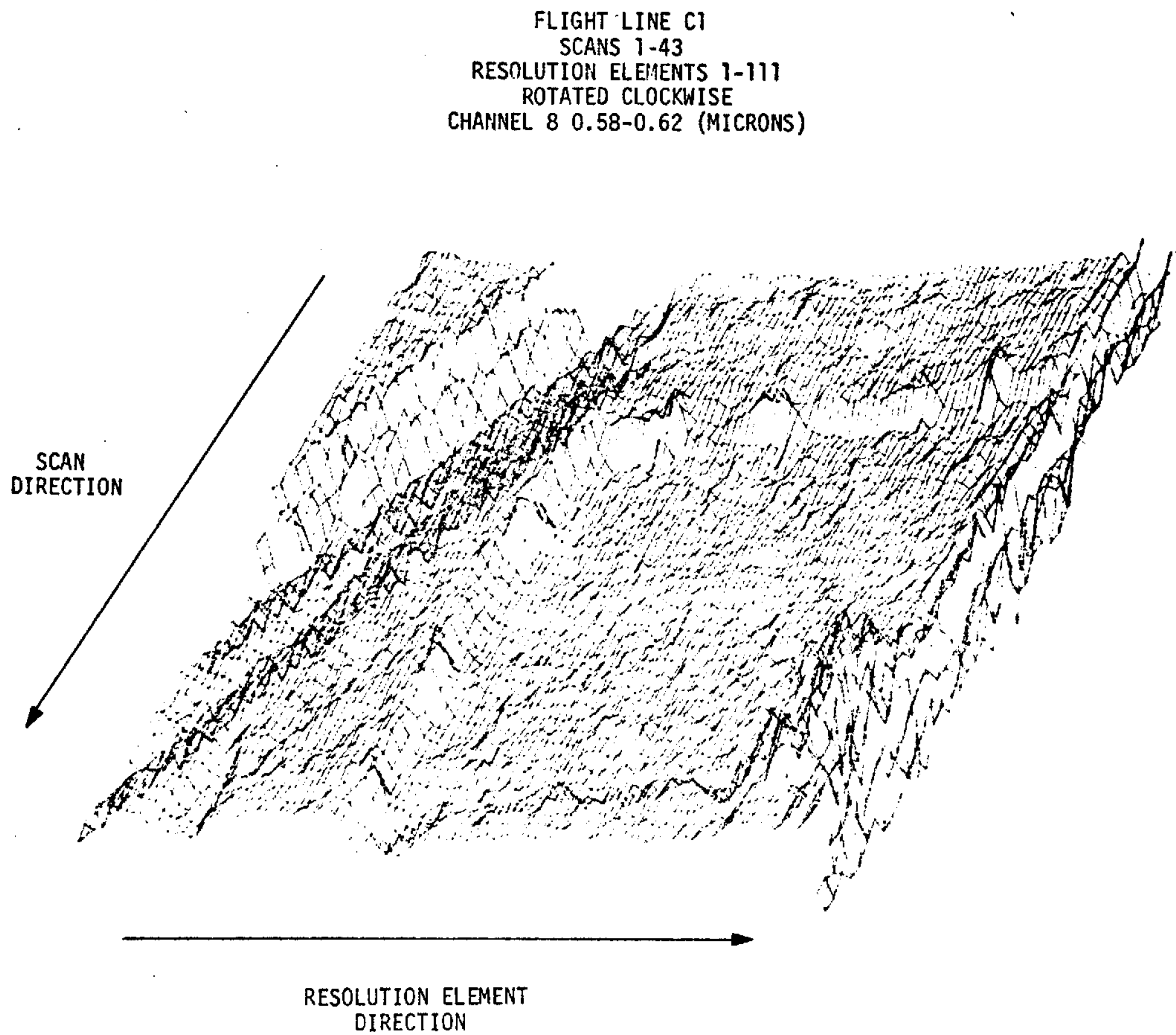


Figure 4 Flight Data Display

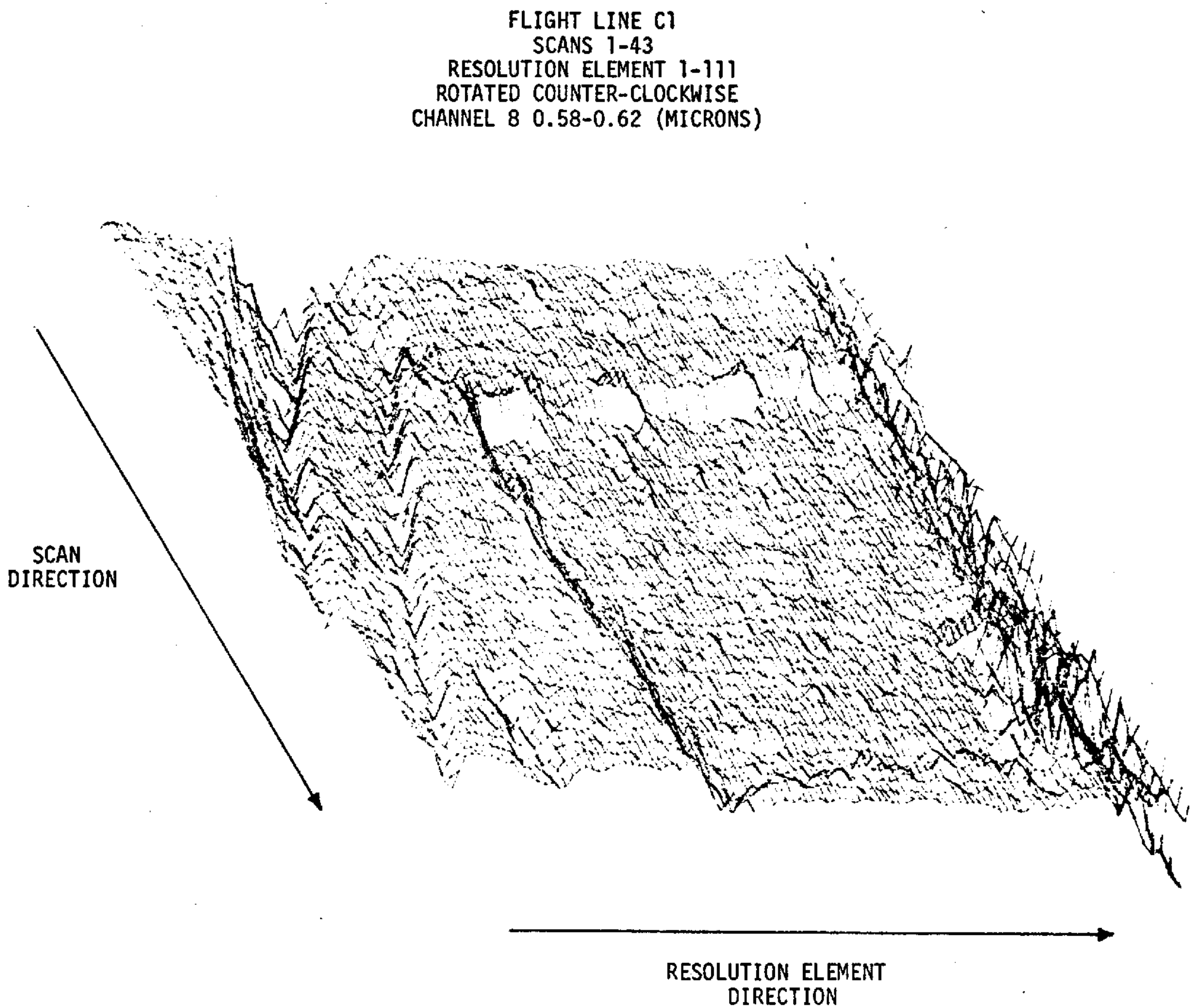


Figure 5 Flight Data Display

FLIGHT LINE C1
SCANS 1-43
RESOLUTION ELEMENTS 1-111
ROTATED CLOCKWISE
CHANNEL 2 0.44-0.46 (MICRONS)

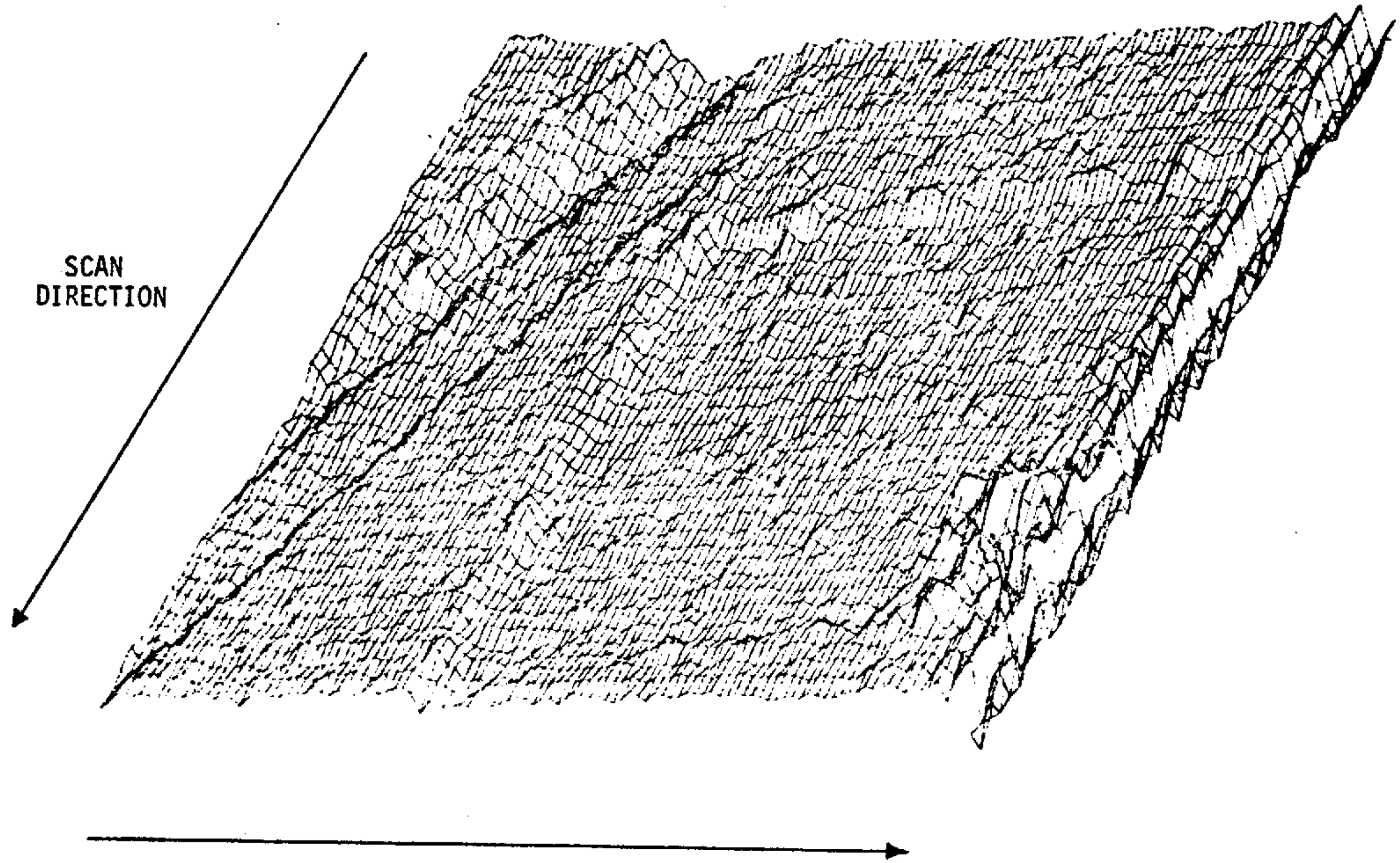


Figure 6 Flight Data Display

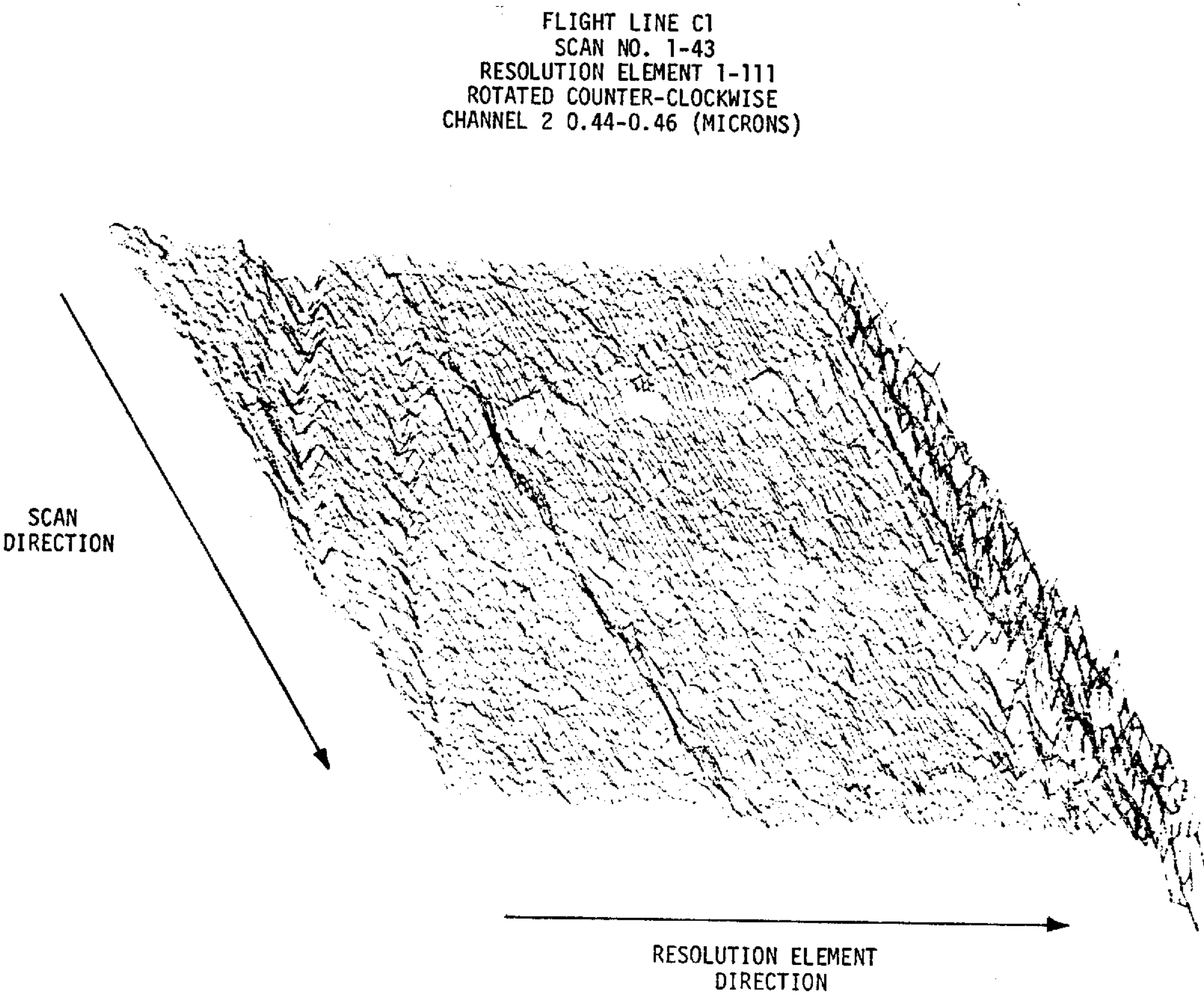


Figure 7 Flight Data Display

A new technique has been employed, that adapts itself to data of any nature, particularly cluster data where the data clusters are tremendously scattered. The technique requires only as much storage as actually required by the number of different data pairs and the size of the data storage required is not dependent on the ranges of the data.

6. JOINT PROBABILITY DENSITY FUNCTION

A joint probability density function program is a necessary tool in the analysis of earth resources flight data. This program gives a preliminary insight to the commonality of occurrences observed from two time-series data traces.

For applications to feature extraction, the outputs of this program give insight to the distinctness of different features and the location of decision boundaries necessary to separate different features.

The joint probability distribution is calculated digitally by selecting a range [a, b] on both sets of data X, Y and dividing the range into K intervals. This gives

$$C = \frac{b-a}{K} = \text{scale factor}$$

To find the X and Y coordinate for each variable X and Y jointly, an indexing pointer is calculated for each value of X and Y by

$$I_x = \frac{(x_n-a)}{C} \quad \text{and} \quad I_y = \frac{(y_n-a)}{C}$$

An integer one is added to the coordinate $p(I_x,I_y)$ for every value of X and Y in the interval [a,b]. The coordinate $p(I_x,I_y)$ contains the total number of occurrences.

A. Computer Program

Data points are paired (user option), given a class, and are stored in two single arrays, one for the X coordinates, NP_x, and one for the Y coordinates, NP_y. The number of data pairs that are common is accumulated in an array NKNT (i) identifying the class and the number of occurrences, i, of that class.

The first data pair read from the input tape, will be assigned class number 1 automatically. All subsequent data pairs read from the input tape, will be compared with all the individual data pairs that have been assigned a particular class. If no match is found during the comparison, a new class is created, using this data pair. There can be up to 4000 different classes of paired data points stored.

Once all the data pairs are classified, the data pairs are arranged and sorted such that the Y coordinates or vertical data are sorted in descending order and the X coordinates or horizontal data are sorted in ascending order. Data are rearranged in such a way that the largest spread between the minimum and maximum is displayed on the vertical axis, which is printed down the print paper. An example of the three data arrays containing the joint probability before and after the vertical sort is shown in Figure 8a and 8b.

B. Core Storage Image

For any data configuration, the core storage image will appear as in Figure 9a. Projecting this onto an NXM storage array used in the conventional method, shows the wasted storage which is used to store blanks (or no occurrences). See Figure 9b. The conventional data storage method required for this example is 6 x 7, or 42 core storage locations. The present scheme requires only 21 core storage locations to display the above example. The advantage, also, is that only 21 core storage locations are necessary regardless of the scatter of the joint probabilities.

$NP_x(8)_1$	$NP_y(6)_1$	$NKNT(1)_1$
• $(6)_2$	• $(12)_2$	• $(6)_2$
• $(12)_3$	• $(8)_3$	• $(8)_3$
• $(8)_4$	• $(7)_4$	• $(12)_4$
$(2)_5$	$(7)_5$	$(8)_5$
$(14)_6$	$(5)_6$	$(7)_6$
$(2)_7$	$(3)_7$	$(6)_7$

a. Before Vertical Sort

$NP_x(6)_1$	$NP_y(12)_1$	$NKNT(6)_1$
• $(12)_2$	• $(8)_2$	• $(8)_2$
• $(8)_3$	• $(7)_3$	• $(12)_3$
• $(2)_4$	• $(7)_4$	• $(8)_4$
$(8)_5$	$(6)_5$	$(1)_5$
$(14)_6$	$(5)_6$	$(7)_6$
$(2)_7$	$(3)_7$	$(6)_7$

b. After Vertical Sort

Figure 8 Data Arrays Before and After Vertical Sort

$NP_x(4)_1$	$NP_y(7)_1$	$NKNT(6)_1$
• $(3)_2$	• $(6)_2$	• $(8)_2$
• $(6)_3$	• $(5)_3$	• $(12)_3$
• $(2)_4$	• $(5)_4$	• $(8)_4$
$(6)_5$	$(3)_5$	$(1)_5$
$(5)_6$	$(3)_6$	$(7)_6$
$(1)_7$	$(2)_7$	$(6)_7$

a. Core Storage Configuration

•	•	•	6	•	•
•	•	8	•	•	•
•	8	•	•	•	12
•	•	•	•	•	•
•	•	•	•	7	1
6	•	•	•	•	•
•	•	•	•	•	•

b. Conventional Core Storage Configuration

Figure 9 Core Storage Image

In Figure 10a and 10b is an example of the same number of paired classes, but the data pairs are more widely scattered. In this example, the conventional data storage required is 14 x 12 or 168 core locations. The present scheme still required only 21 core locations.

C. Output Display

To display the arrays as a joint probability distribution, the first location in the NP_y array is examined and a decremental counter is set equal to this value. All the data points in the NP_x array that are paired with this NP_y value, are collected along with their number of occurrences NKNT(i), and stored in a working array. This working array is sorted in ascending order for printing from left to right across the print paper. A print line is loaded with blank characters and the blank character is replaced with the number of occurrences in that location, if any exist. The number of occurrences is designated by an alphanumeric character of some hierarchy ordered by the user. The print line is then output to the printer. The next location in the NP_y array is examined and the counter is decreased by one. If the NP_y value is less than the counter, a print line filled with blank characters is printed and tagged with the value of the counter. The counter is then decremented and compared with the NP_y value again. If there is a comparison, then the above procedure is repeated. This continues until the NP_y array is exhausted.

D. Data Problem Parameters

NCH	Total number of channels on input tape
NSPS	Samples per logical record or scan line
NSCANS	Number of logical records or scan lines to process
NSKIP	Initial physical records to skip before processing
NSTART	Starting sample number

$NP_x(6)_1$	$NP_y(12)_1$	$NKNT(6)_1$
• $(12)_2$	• $(8)_2$	• $(8)_2$
• $(8)_3$	• $(7)_3$	• $(12)_3$
• $(2)_4$	• $(7)_4$	• $(8)_4$
$(8)_5$	$(6)_5$	$(1)_5$
$(14)_6$	$(5)_6$	$(7)_6$
$(2)_7$	$(3)_7$	$(6)_7$

a. Core Storage Configuration

•	•	•	•	•	6	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	8	•	•
•	8	•	•	•	•	•	12	•	•	•	•	•	•
•	•	•	•	•	•	•	1	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	7
•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	6	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•

b. Conventional Core Storage Configuration

Figure 10 Core Storage Image

NSTOP	Stopping sample number
LTN	Logical tape unit to load input data
NOJP	Number of paired joint probabilities to process
IMX	X channel select
IMY	Y channel select
SCALE	Used to scale data
BIAS	Used to shift data

7. BOUNDARY MAPPING

A concerted effort has been made to improve boundary mapping techniques and to extend these ideas and techniques in order to classify different homogeneous areas of ground scene data. This section briefly describes one of several boundary enhancement techniques, which were investigated, and contains examples of test cases and problem parameter inputs.

The technique employed in this computer program, incorporates a moving rectangle made up of four adjacent resolution elements moved successively through the data. The configuration, produced by the magnitude of any four adjacent resolution elements connected by an imaginary path, will be defined as the area. Isometric displays of the ground scene (Figure 11) show different area configurations produced by any four adjacent resolution elements. The possible area configuration models are shown in Figure 12. From these configurations, the equations for calculating the area were derived, based on two cases presented by the data. Case I is the area produced by four adjacent resolution elements along the Y and Z plane, and case II is the area produced by four adjacent resolution elements along the X and Z plane. The equations that were derived (A_1, B_1, C_1 for case I and A_2, B_2, C_2 for case II) are shown in Figure 13a.

Several equations that appear here are redundant and are eliminated by combining similar equations. The composite test statements that determine which equation to use, are shown in Figure 13b, and a flow diagram of the decision logic incorporated in the computer program, is shown as Figure 14.

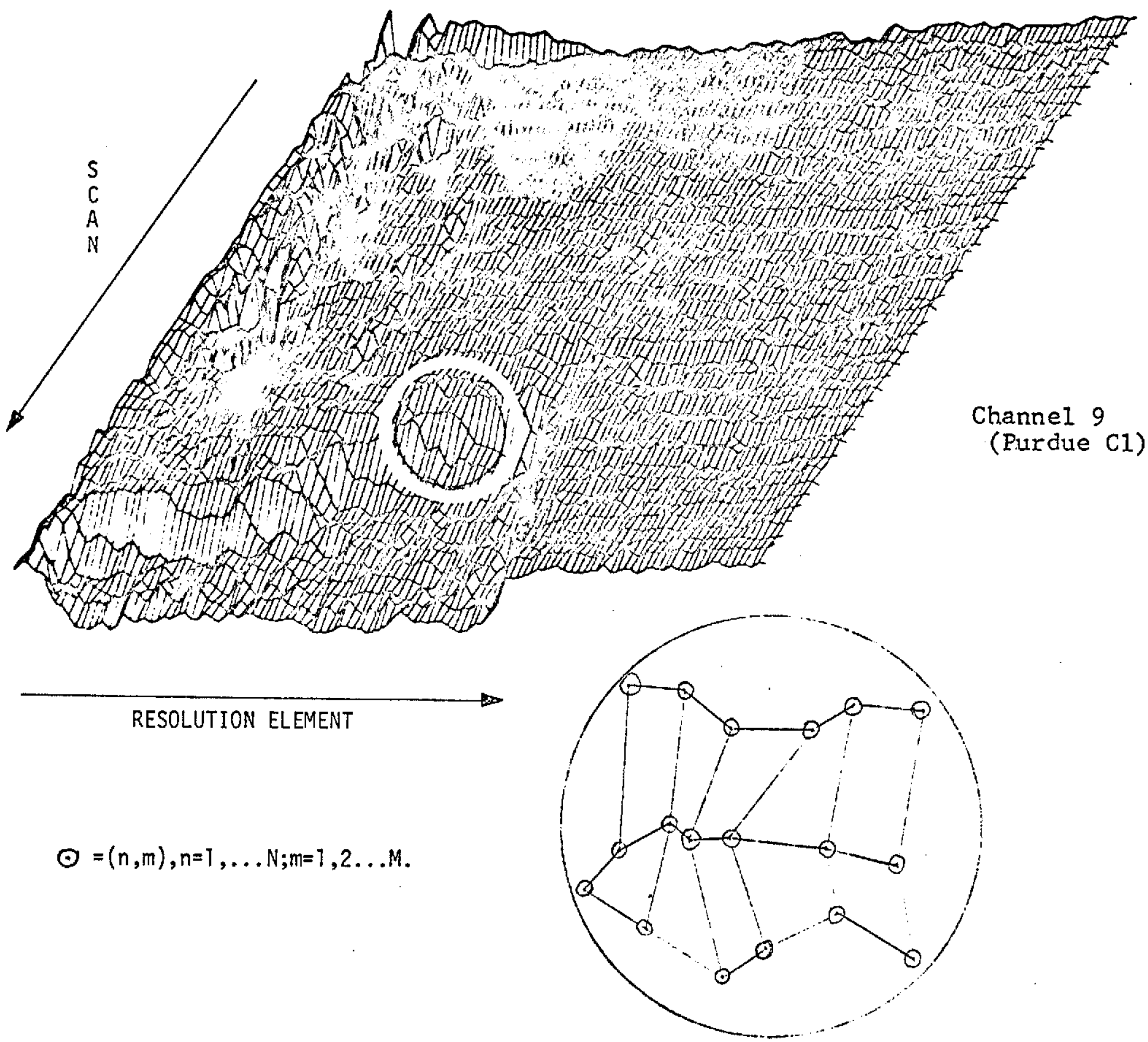


Figure 11 Isometric Display of Ground Scene

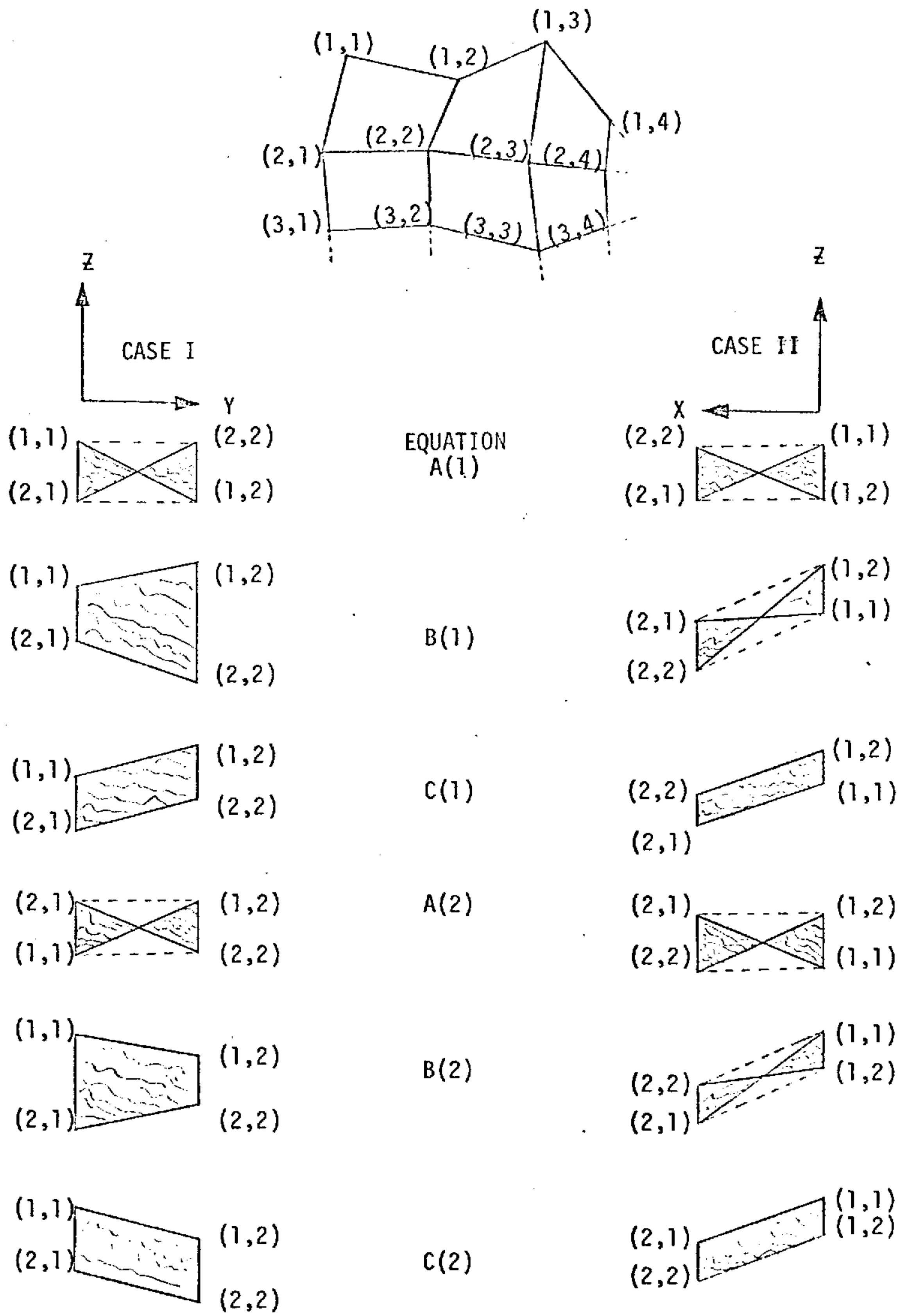


Figure 12 Area Configurations Produced by Four Adjacent Data Samples

CASE I

$$\tfrac{1}{4}[(X_n,Y_n)-(X_n,Y_{n+1})-(X_{n+1},Y_n)+(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{2}[(X_n,Y_n)-(X_n,Y_{n+1})+(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{2}[(X_n,Y_n)+(X_n,Y_{n+1})-(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{4}[-(X_n,Y_n)+(X_n,Y_{n+1})+(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{2}[(X_n,Y_n)+(X_n,Y_{n+1})-(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{2}[(X_n,Y_n)+(X_n,Y_{n+1})-(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$

CASE II

$$\tfrac{1}{4}[(X_n,Y_n)-(X_n,Y_{n+1})-(X_{n+1},Y_n)+(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{4}[(X_n,Y_n)-(X_n,Y_{n+1})+(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{2}[-(X_n,Y_n)+(X_n,Y_{n+1})-(X_{n+1},Y_n)+(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{4}[-(X_n,Y_n)+(X_n,Y_{n+1})+(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})$$
$$\tfrac{1}{4}[(X_n,Y_n)-(X_n,Y_{n+1})-(X_{n+1},Y_n)+(X_{n+1},Y_{n+1})]$$
$$\tfrac{1}{2}[(X_n,Y_n)-(X_n,Y_{n+1})+(X_{n+1},Y_n)-(X_{n+1},Y_{n+1})]$$

(a)

COMPOSITE TEST STATEMENTS

$$\left.\begin{array}{l} X_n,Y_n > X_n,Y_{n+1} \\ X_{n+1},Y_n < X_{n+1},Y_{n+1} \\ X_{n+1},Y_{n+1} > X_n,Y_{n+1} \end{array}\right] \text{— A(1)}$$

$$\text{A(2) —}\left[\begin{array}{l} X_n,Y_n < X_n,Y_{n+1} \\ X_{n+1},Y_n > X_{n+1},Y_{n+1} \\ X_n,Y_n < X_{n+1},Y_n \end{array}\right.$$

$$\left.\begin{array}{l} X_n,Y_n < X_n,Y_{n+1} \\ X_{n+1},Y_n > X_{n+1},Y_{n+1} \\ X_n,Y_n > X_{n+1},Y_n \end{array}\right] \text{— B(1)}$$

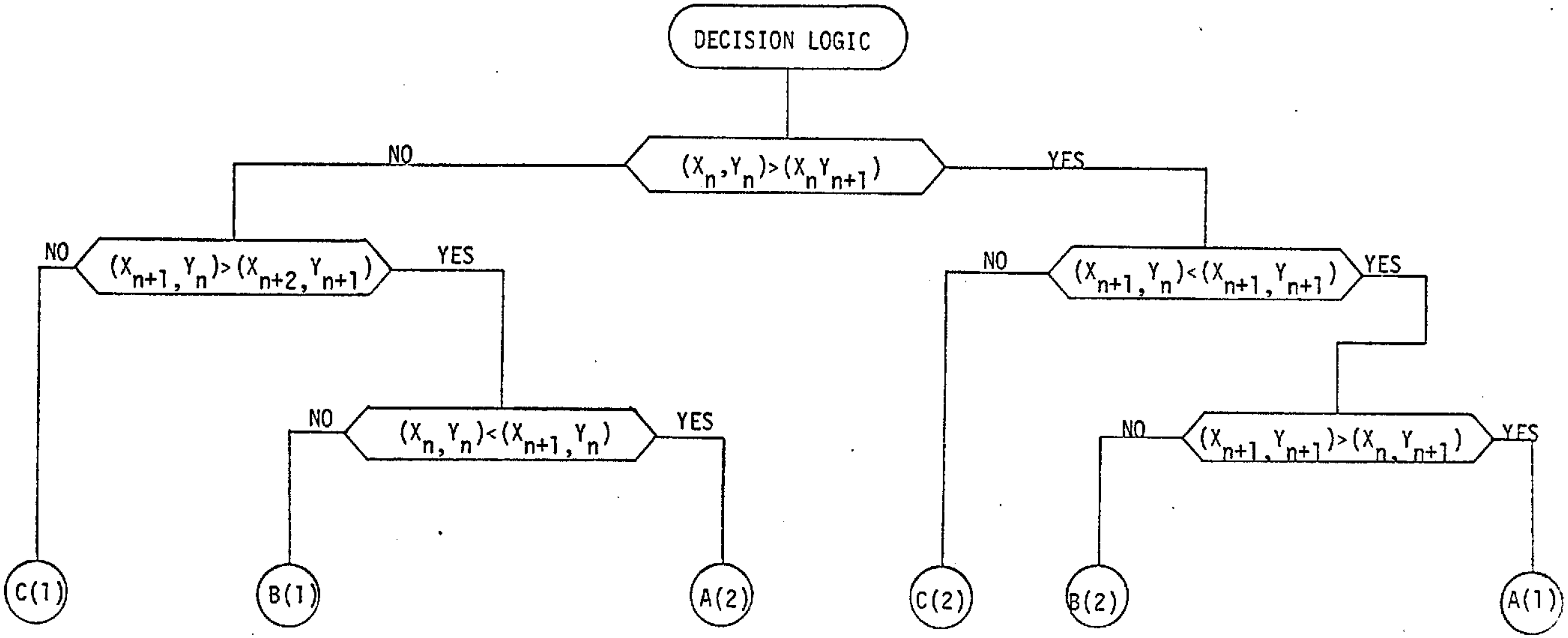
$$\text{B(2) —}\left[\begin{array}{l} X_n,Y_n > X_n,Y_{n+1} \\ X_{n+1},Y_n < X_{n+1},Y_{n+1} \\ X_{n+1},Y_{n+1} < X_n,Y_{n+1} \end{array}\right.$$

$$\left.\begin{array}{l} X_n,Y_n < X_n,Y_{n+1} \\ X_{n+1},Y_n < X_{n+1},Y_{n+1} \end{array}\right] \text{— C(1)}$$

$$\text{C(2) —}\left[\begin{array}{l} X_n,Y_n > X_n,Y_{n+1} \\ X_{n+1},Y_n > X_{n+1},Y_{n+1} \end{array}\right.$$

(b)

Figure 13 Area Equations for Four Adjacent Data Samples



$A(1)_H = \frac{1}{2}[(X_n, Y_n) - (X_n, Y_{n+1}) - (X_{n+1}, Y_n) + (X_{n+1}, Y_{n+1})]$
$A(1)_V = A(1)_H$

$B(1)_H = \frac{1}{2}[(X_n, Y_n) - (X_n, Y_{n+1}) + (X_{n+1}, Y_n) - (X_{n+1}, Y_{n+1})]$
$B(1)_V = \frac{1}{2} B(1)_H$

$C(1)_H = \frac{1}{2}[(X_n, Y_n) + (X_n, Y_{n+1}) - (X_{n+1}, Y_n) - (X_{n+1}, Y_{n+1})]$
$C(1)_V = \frac{1}{2}[-(X_n, Y_n) + (X_n, Y_{n+1}) - (X_{n+1}, Y_n) + (X_{n+1}, Y_{n+1})]$

$A(2)_H = \frac{1}{2}[-(X_n, Y_n) + (X_n, Y_{n+1}) + (X_{n+1}, Y_n) - (X_{n+1}, Y_{n+1})]$
$A(2)_V = A(2)_H$

$B(2)_H = \frac{1}{2}[(X_n Y_n) + (X_n, Y_{n+1}) - (X_{n+1}, Y_n) - (X_{n+1}, Y_{n+1})]$
$B(2)_V = \frac{1}{2}[(X_n, Y_n) - (X_n, Y_{n+1}) - (X_{n+1}, Y_n) + (X_{n+1}, Y_{n+1})]$

$C(2)_H = \frac{1}{2}[(X_n, Y_n) + (X_n, Y_{n+1}) - (X_{n+1}, Y_n) - (X_{n+1}, Y_{n+1})]$
$C(2)_V = \frac{1}{2}[(X_n, Y_n) - (X_n, Y_{n+1}) + (X_{n+1}, Y_n) - (X_{n+1}, Y_{n+1})]$

* H = Area in Horizontal   V = Area in Vertical Direction

Figure 14 Flow Diagram of Decision Logic

A probability density distribution of the area is calculated for each channel of data and every scan. An estimate of this probability density function is obtained digitally by dividing the range for X into an appropriate number of class intervals, say d_i, and by tabulating the number of occurrences in each of these intervals. The number of occurrences N_i in each interval satisfies the equation

$$N = \sum_{i=0}^{k-1} N_i$$

where k = number of class intervals and N is the total population.

This number sequence $\{N_i\}$ is found by

$$N_1 = \text{Total number of X:} \quad X \leq d_1$$

$$N_2 = \text{Total number of X:} \quad d_1 < X \leq d_2$$

$$N_3 = \text{Total number of X:} \quad d_2 < X \leq d_3$$

$$\vdots$$

$$N_k = \text{Total number of X:} \quad d_{k-1} < X \leq d_k$$

A plot of the probability distribution of channel 3 and channel 12, both horizontal and vertical, are shown in Figures 15 and 16. These plots reflect the distribution of one scan of data, in particular, scan number 5. The criterion to determine if a resolution element reflects a change in a homogeneous population of ground scene is the mean of the probability distribution of the area plus its associated σ. All area calculations of four adjacent resolution elements that fall outside of the mean plus σ will be flagged as a boundary.

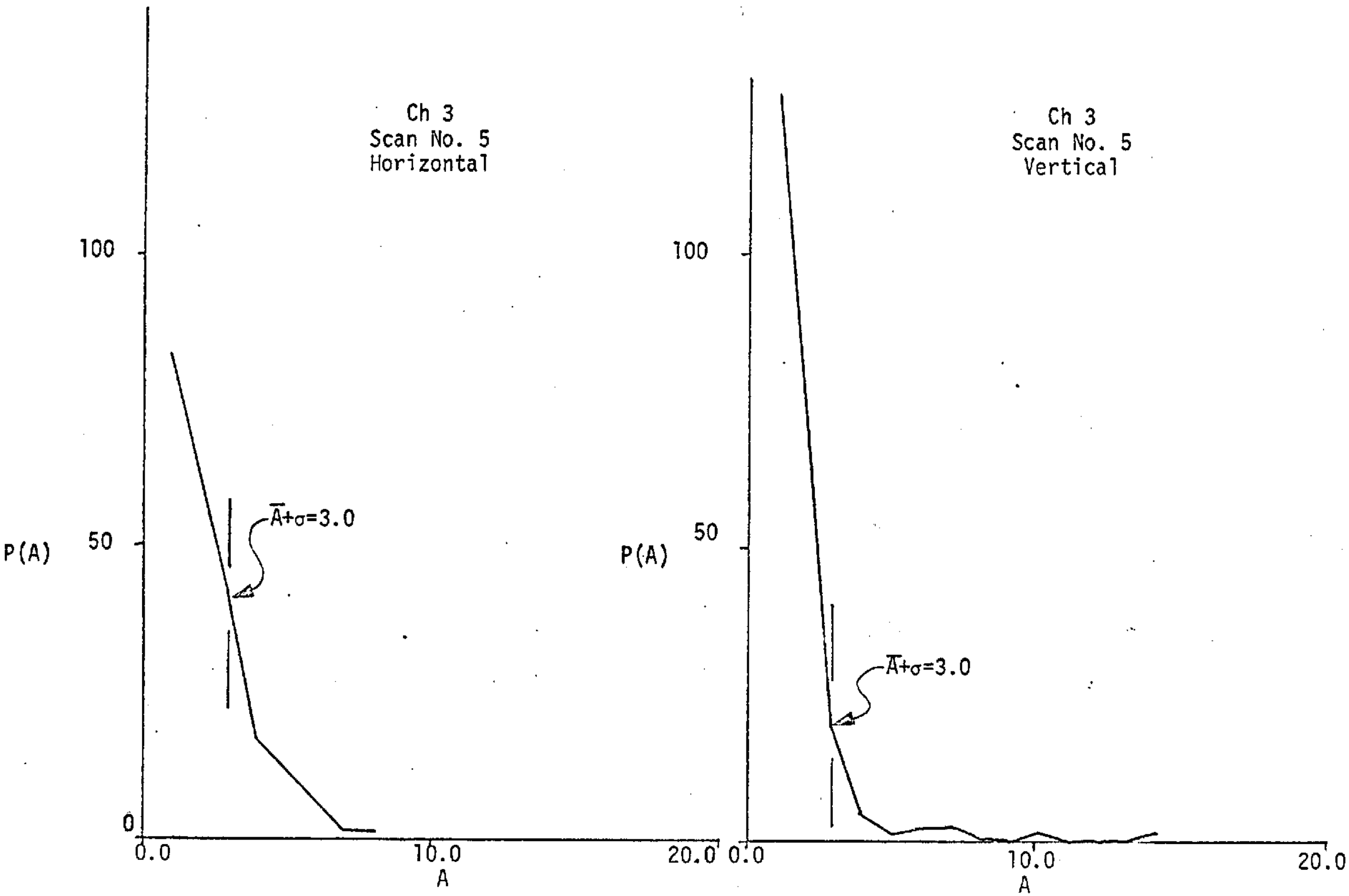


Figure 15 Boundary Determination, Probability Distribution Channel 3

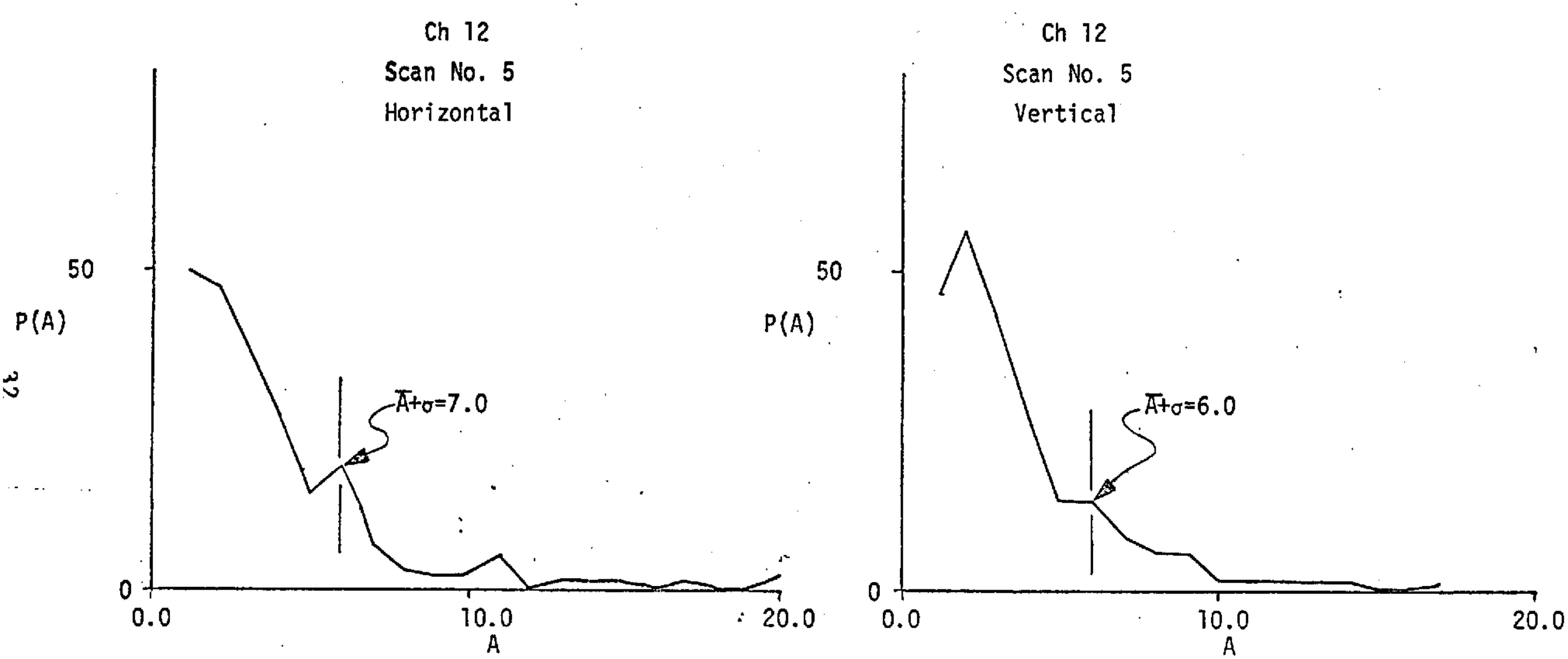


Figure 16 Boundary Determination, Probability Distribution Channel 12

The mean area of the probability distribution, P(A), is calculated by

$$\overline{P(A)} = \frac{\sum_i P(A_i)\ A_i}{\sum_i P(A_i)} \qquad i=1,2,\ldots MAX$$

on σ by

$$\sigma = \left\{ \frac{\sum_i P(A_i)\ A_i^2}{\sum_i P(A_i)} - \left(\overline{P(A)}\right)^2 \right\}^{1/2} \qquad i=1,2\ldots MAX$$

In the test cases, all 12 channels were used, except for two cases where channel 3 and channel 12 were run separately, and boundary flags set for each channel. The final decision for determining the boundary, is based on an input parameter selected by the user. The number of channels indicating a boundary has to be greater than this value. In Figure 17 only channel 12 was used. This produced too many boundaries and the separation of different homogeneous areas was not clearly defined. In Figure 18 only channel 3 was used; this produced too few boundaries. Figure 19 shows all 12 channels with at least seven reflecting boundaries (Input Option); this produced too few boundaries. Figure 20 shows at least three channels of the 12 reflecting boundaries which produced too many boundaries. Figure 21 produces the best boundary map which uses at least four channels of the 12 total.

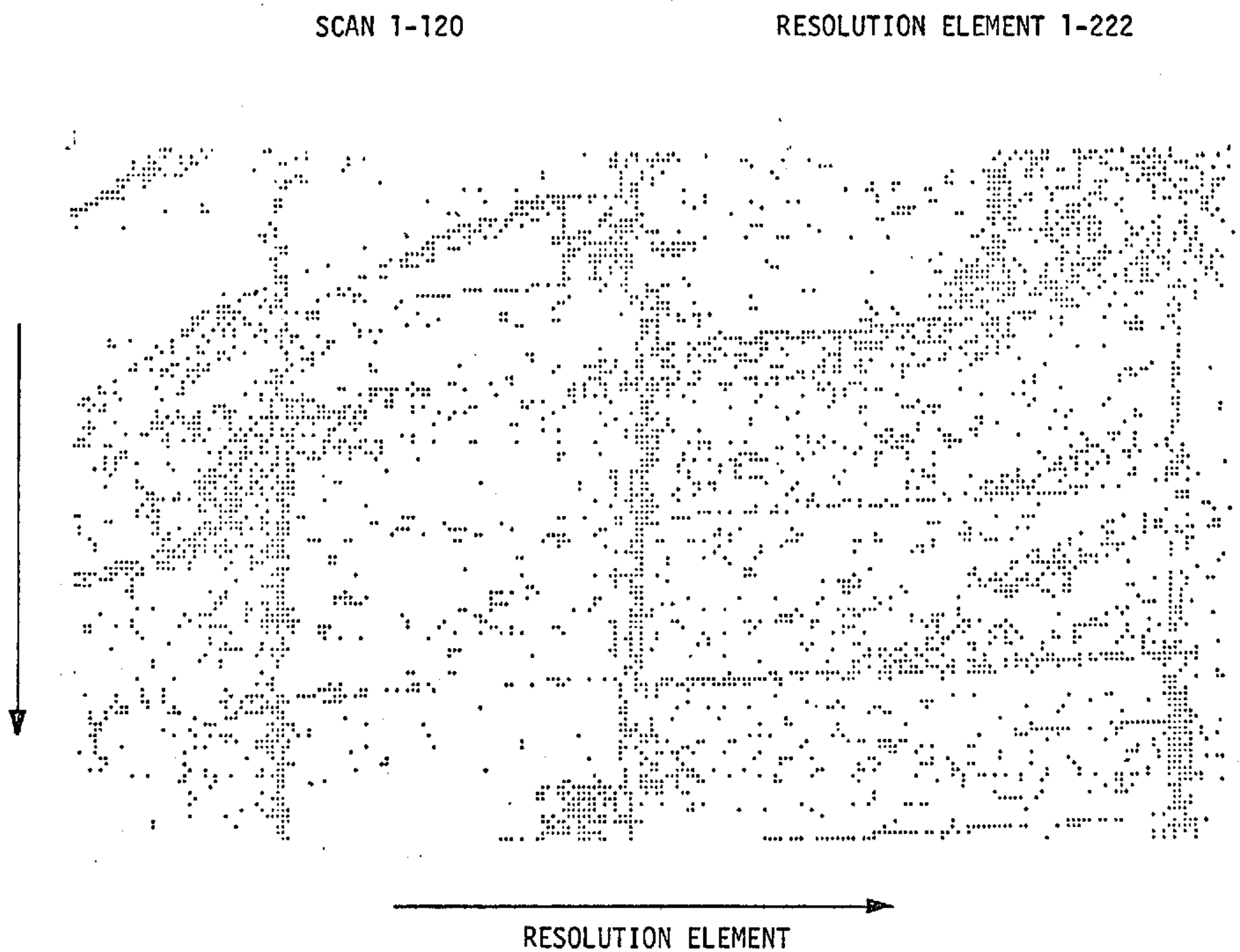


Figure 17 Boundary Determination Channel 12 (Purdue C1)

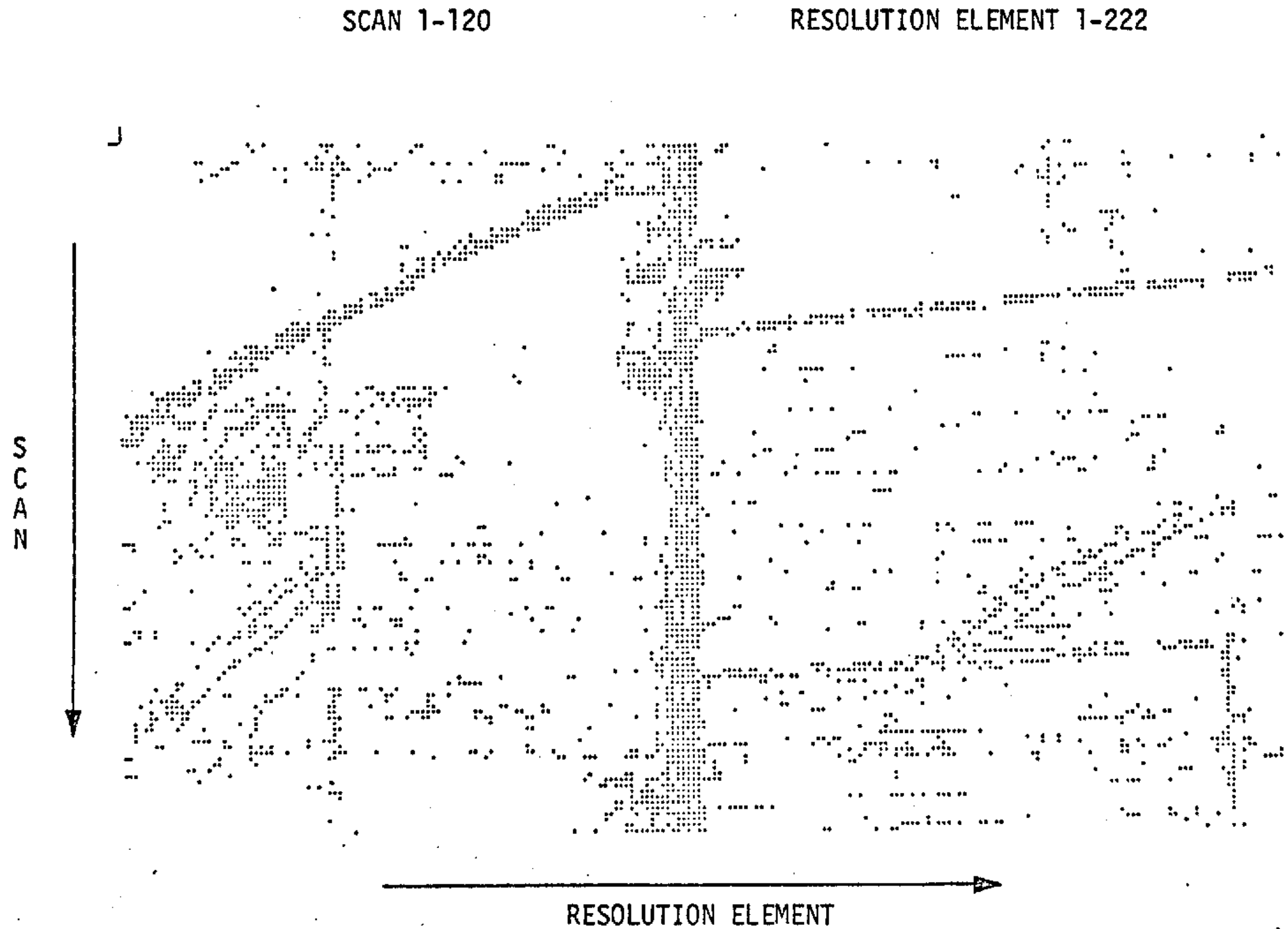


Figure 18 Boundary Determination Channel 3 (Purdue C1)

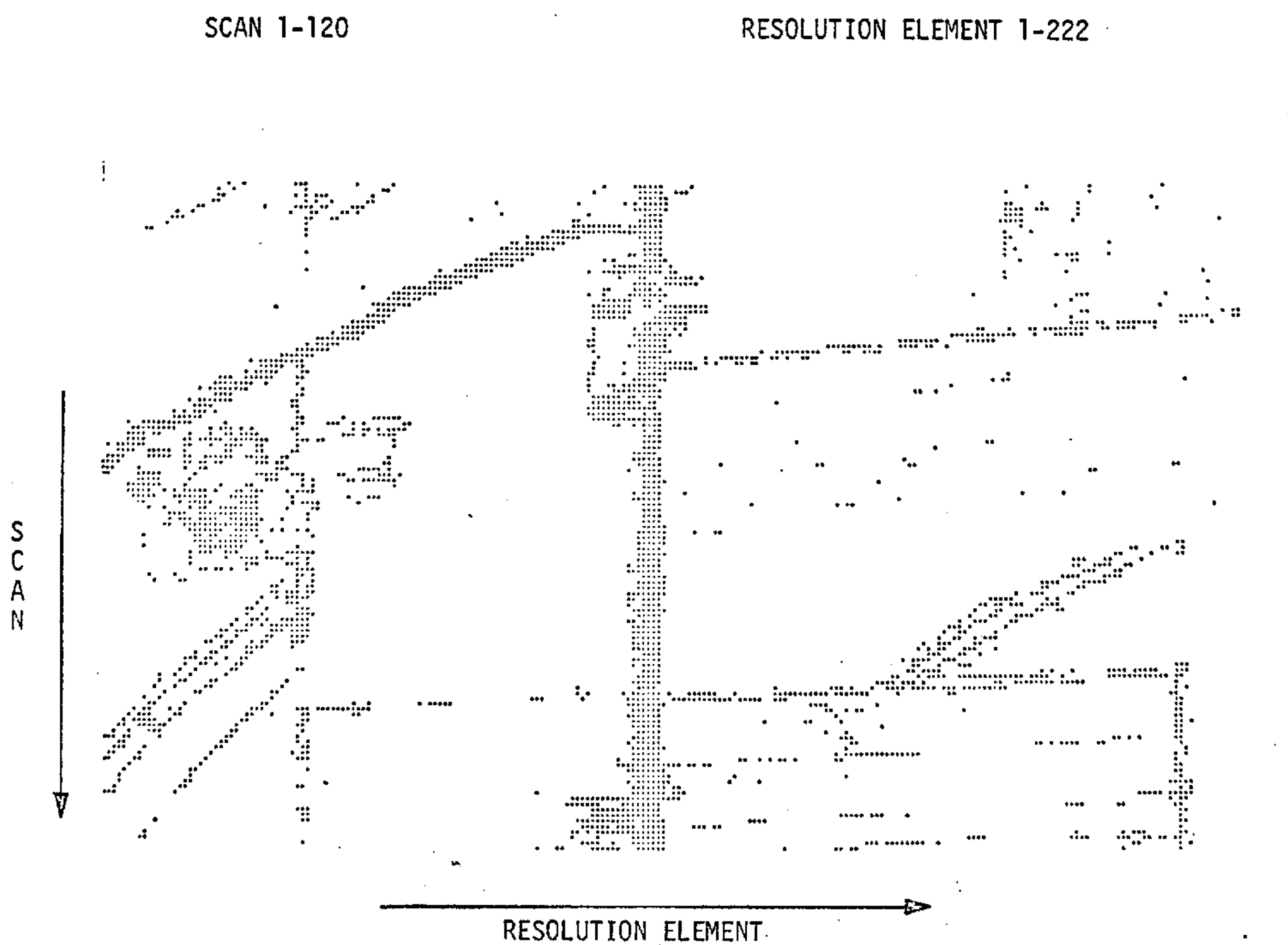


Figure 19 Boundary Determination Channels 7 of 12 (Purdue C1)

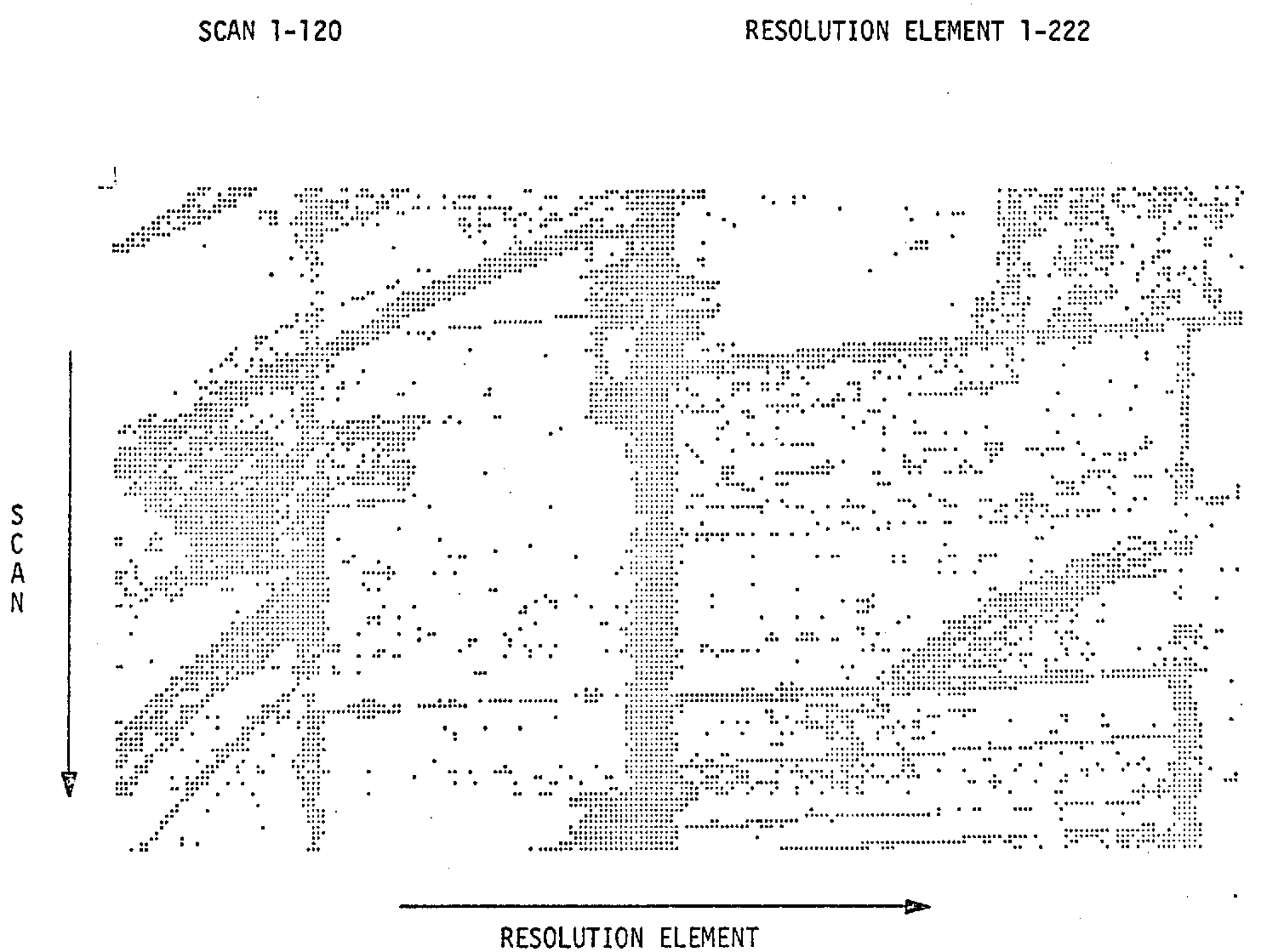


Figure 20 Boundary Determination Channels 3 of 12 (Purdue C1)

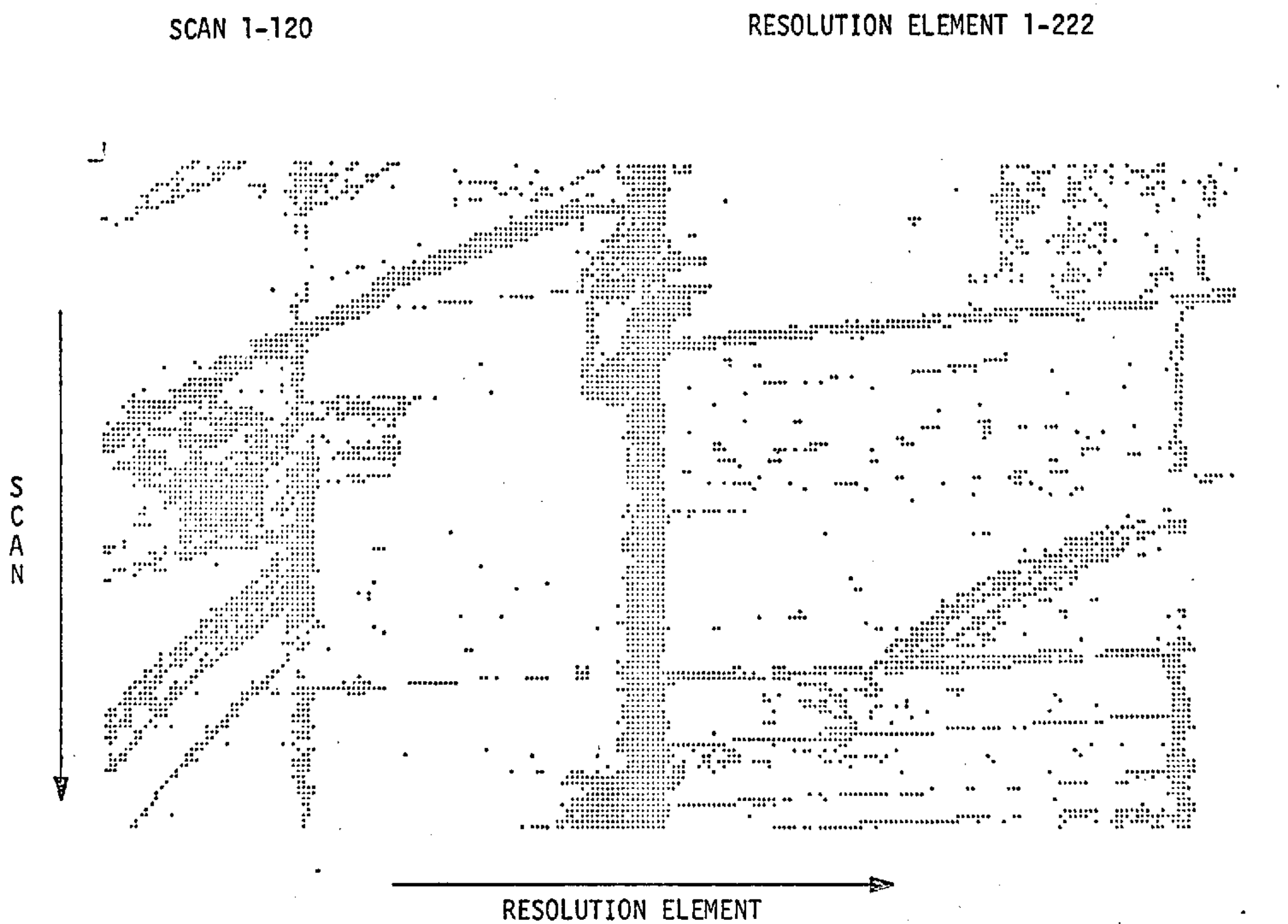


Figure 21 Boundary Determination Channels 4 of 12 (Purdue C1)

A. Data Problem Parameters

NSCANS	Number of scans or logical records to process an input tape
NSTART	Starting resolution element in the scan
NSPS	The number of resolution elements or columns to process from each scan ($\leq$ 255)
NCH	Number of channels present on input tape
NVAR	Number of variables or intervals desired in calculating the probability distribution; Maximum of 2000.
NSYM	Number of alphanumeric characters to display boundaries (note: At the present only two characters are used, blanks and dots, no boundary or yes boundary respectively).
ISUM	Total number of channels used in initial boundary calculation
NTEST	Number of channels (plus one), of ISUM, necessary for a final boundary decision
NBTLG	Number of bits per word on the input tape
MODE	Mode of input tape (2=non-FORTRAN, 1=FORTRAN)
NSKIP	Number of records to skip before processing input data
NCRE	Data increment
NBLK	Number of raster counts per character desired in displaying boundary on the Stromberg-Carlson 4020 plotter recorder
INCX	Raster counts incrementation in the X direction
INCY	Same as above in the Y direction
NSTX	Starting position or frame along the X axis
NSTY	Starting position on frame along the Y axis
NWHICH	Channel number desired in calculation of boundary

8. CHANNEL ALIGNMENT

In recent months there has been an increasing demand to process and analyze data collected by array cameras using different filters in the frequency spectrum. In order to analyze data from this camera source, the camera film is digitized and the digitizing process is carried out separately for each filtered film. Gross errors are introduced into the data, mainly missalignment of the film when being digitized by a densitometer. This missalignment must be compensated for before the digitized data can be merged and processed. A process was developed to align the digitized data sets by a matching matrix technique.

Boundaries separating homogeneous areas are calculated from each data set (see section 7) and the boundaries of each data set are correlated and matched. One particular technique proposed was the use of NXM data matrix for each data set where one set is assigned the reference channel and the other sets are correlated with the reference channel by moving the matching matrix successively through the selected data area.

The correlation of the reference channel with other channels is done by matching or correlating a resolution element from the reference channel with all resolution elements in the matching matrix of other channels. If there exists a match or correlation of boundaries, then a cell in the matching matrix display at those coordinates is incremented. The NXM matching matrix is moved by one resolution element, and the next resolution element from the reference channel is matched or correlated with all the resolution elements in the matching matrix of other channels. If there exists a match or correlation of boundaries, then a cell in the matching matrix display at those coordinates is again incremented. This continues as the matching matrix is moved successively through the selected data interval.

An expanding and collapsing technique is employed at the beginning and end of the data set to eliminate edge effects and loss of data. As the NXM matching matrix is moved successively through the data set, the correlation of the two channels, if a correlation exists, is accumulated at the beginning of the data set at the matching matrix display coordinates MATRIX (i,j) going across, where

$$i = \frac{N}{2} - b+1,\ \frac{N}{2} - b+2, \ldots N;\ k = 1,2,3 \ldots \frac{N}{2}\ ,$$

and going down where

$$j = \frac{M}{2} - b+1,\ \frac{M}{2} - b+2, \ldots M;\ k = 1,2.3 \ldots \frac{M}{2}\ .$$

This continues until the entire matching matrix is within the objective data, thus eliminating the edge effect. Once the matching matrix is within the objective data then i = 1,2,3...N; and j = 1,2,3...M. Upon termination of the matching matrix at the end of a data set, a collapsing technique is employed. This technique is the same as the expanding technique except the accumulation at the end of the data set at the matching matrix display coordinates MATRIX (j,i) is $i = 1,2.3 \ldots N-k$, where $k = 1,2,3 \ldots \frac{N}{2}$; and $j = 1,2,3 \ldots M-k$, where $k = 1,2.3 \ldots \frac{M}{2}$.

The matching matrix is then printed for each combination of correlations. No boundary and boundary correlation of two data channels (0.1), and boundary and boundary correlation of two data channels (1,1). Alignment shifts can be performed by locating the peaks in the NXM matrix where boundaries of two channels are matched. The matching matrix is shown in the output example. This peak occurred at coordinates (-5, +4), therefore, the channel being correlated with the reference should be shifted -5 data records and +4 resolution elements to align the two channels.

This technique corrects for any horizontal and vertical alignment satisfactorily, however it does not correct for any skewness introduced, while the film is being digitized, or any distortion in the camera array angles. A technique that fully corrects for this type of missalignment is presently being developed, but due to inadequate computer turnaround time, results are not available.

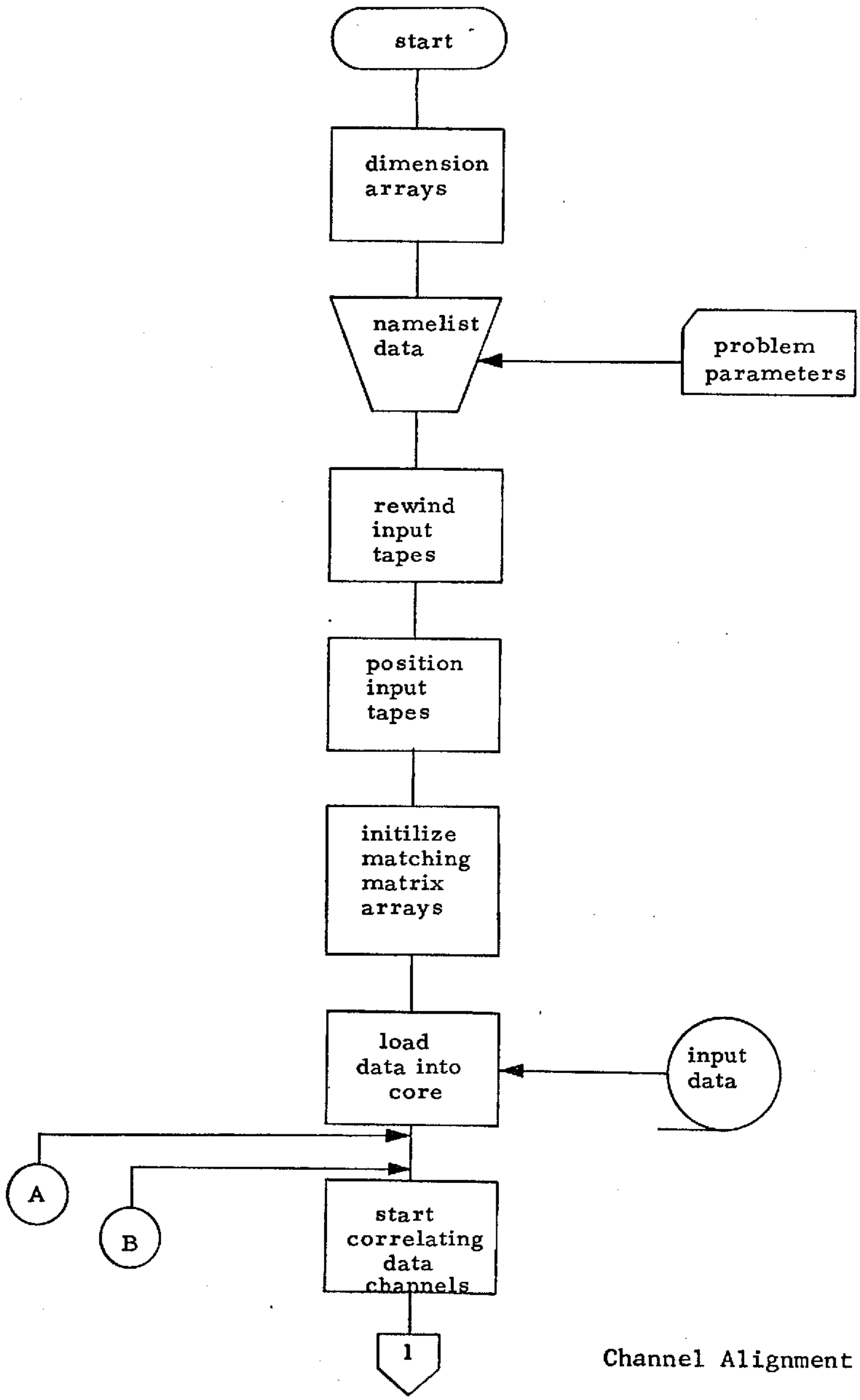
A. Computer Program

A program was written to match and correlate up to three data channels and as many different intervals of a data set as desired. This will enable the user to calculate the alignment in areas of the data set where boundaries are clear and can be detected. Input to the program is three FORTRAN formatted fixed point binary tapes containing boundary flags (fixed point integer 1 for boundary and 0 for areas that are homogeneous).

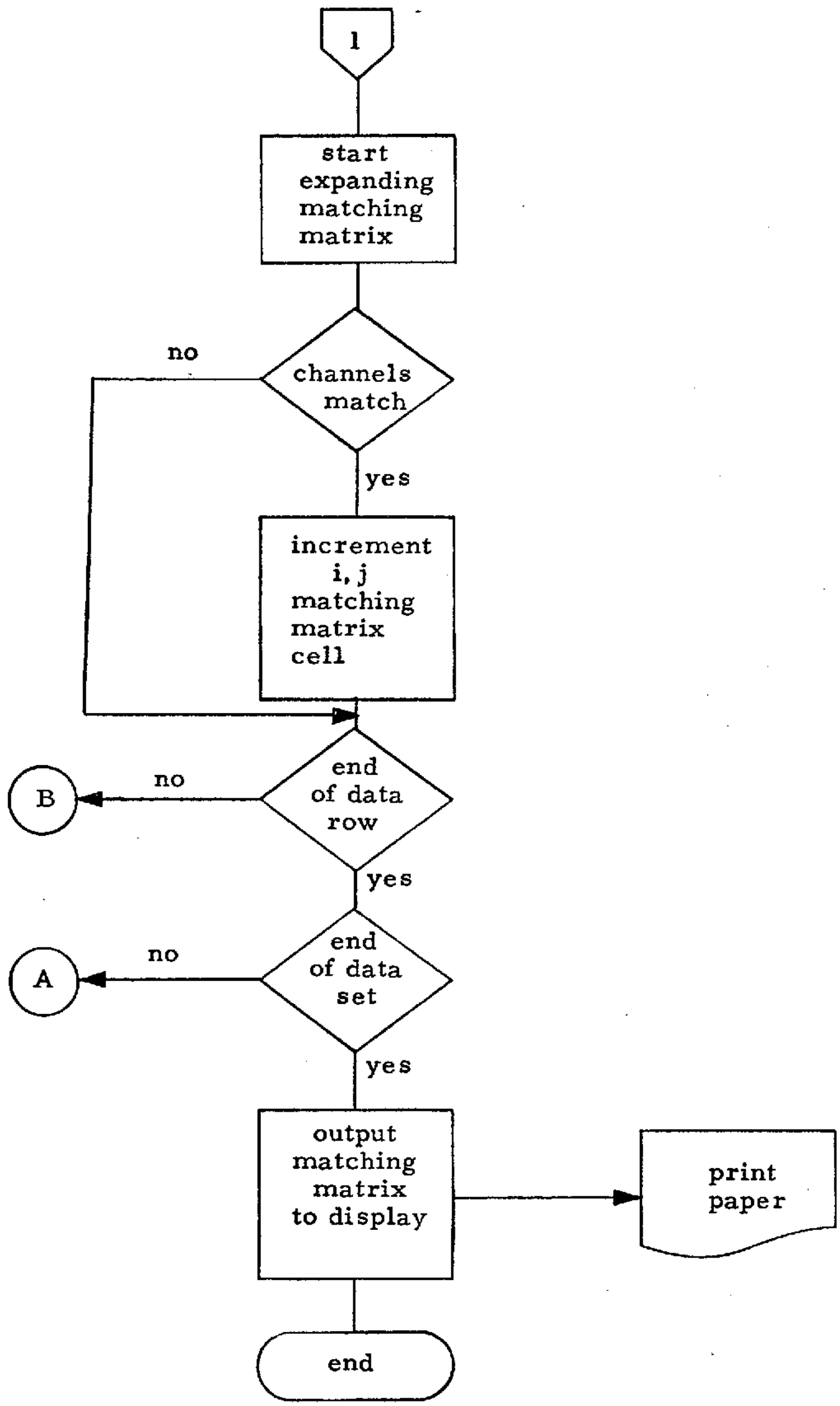
B. Data Problem Parameters

NTIMES	Number of sections to correlate
LT9	Logical unit to load input tape number one
LT10	Logical unit to load input tape number two
LT11	Logical unit to load input tape number three
NSTART	Starting resolution element
NSTOP	Final resolution element
NSCANS	Number of scans in the matching matrix
NSPS	Number of resolution elements across the scan in the matching matrix
NREFX	Starting resolution element in the reference channel; Tape number one.
NREFY	Starting scan number in the reference channel; Tape number one.
NCH1X	Starting resolution element in the channel to be correlated; Tape number two.
NCH1Y	Starting scan in the channel to be correlated; Tape number two.
NCH2X	Starting resolution element in the channel to be correlated; Tape number three.
NCH2Y	Starting scan in the channel to be correlated; Tape number three.

C. Program Flow Chart



Channel Alignment



Channel Alignment (Concluded)

OUTPUT EXAMPLE

MATCHING MATRIX (1 AND 1)

CH 2 WITH CH 1

174	176	176	182	192	160	170	193	198	192	167	182	198	196	159	176	168	161	169	181	156	171	171	165	149	146	167
171	178	172	183	185	164	169	185	193	210	189	173	181	196	187	178	175	171	190	176	179	165	178	187	157	158	173
177	179	178	185	204	174	170	184	208	212	197	183	184	184	183	168	177	177	178	184	189	168	171	173	168	168	179
189	191	183	193	204	197	185	193	205	201	195	199	186	182	190	184	180	178	178	192	196	169	163	187	176	162	176
182	179	208	192	202	218	208	194	202	227	212	197	214	209	192	193	198	186	181	201	193	173	182	195	171	179	185
172	204	213	193	191	196	196	202	199	218	212	202	215	216	198	218	227	188	189	192	199	196	201	191	174	184	190
186	185	201	189	198	194	212	196	198	230	221	225	231	216	216	209	217	200	204	195	187	193	197	200	175	185	167
193	167	199	187	175	183	191	215	216	236	232	214	226	245	213	206	216	197	197	185	177	190	165	193	188	181	178
163	184	200	171	189	190	195	207	219	237	220	221	232	214	221	226	212	211	211	197	198	205	205	193	180	196	182
208	187	193	183	194	204	202	211	249	238	240	226	233	232	223	236	216	197	212	207	206	222	209	211	210	197	191
210	194	188	186	192	218	204	218	$\boxed{256}$	239	249	250	225	204	217	234	213	192	192	212	220	200	203	209	200	209	198
207	206	189	197	195	218	200	207	243	236	246	243	234	233	224	234	200	193	199	219	220	203	204	206	192	211	195
198	222	230	186	194	204	197	212	241	235	232	223	237	236	236	227	216	186	195	203	217	216	210	218	190	210	200
193	196	221	187	205	206	195	225	255	224	236	240	220	226	207	208	212	218	224	211	222	210	219	224	207	198	195
194	183	211	189	194	189	201	219	239	242	244	225	230	243	206	216	209	214	212	216	227	232	233	214	208	213	175
184	164	196	192	192	187	208	210	215	238	239	222	232	222	220	211	220	202	190	201	218	230	224	200	202	209	190
183	190	192	186	199	212	196	199	195	225	218	223	197	222	224	226	218	194	210	201	203	228	221	216	209	205	174
176	164	170	188	202	189	168	175	196	222	218	204	201	211	223	217	212	221	231	220	225	228	222	219	224	212	199
177	176	192	173	172	180	165	171	195	212	211	203	191	203	224	227	209	201	201	239	233	230	217	221	219	215	213
183	184	203	187	185	181	169	202	207	211	213	200	195	209	224	234	227	195	198	226	210	167	202	204	206	205	206
181	182	189	194	178	168	181	205	210	224	230	199	190	208	216	213	214	204	217	218	208	204	192	189	219	207	194
159	162	192	187	187	175	175	201	209	219	214	202	193	209	212	202	200	221	210	192	188	214	245	184	204	211	196
170	161	165	173	191	186	191	187	203	210	207	200	218	208	205	224	211	209	217	190	200	204	207	204	197	198	191
181	160	161	174	182	195	195	191	213	215	225	212	217	190	223	226	222	207	215	209	202	194	208	211	184	172	166
152	152	152	191	165	190	180	181	183	226	224	205	214	183	208	222	232	209	200	209	215	196	211	212	210	169	191
144	149	155	173	170	178	207	188	196	217	225	186	198	188	191	200	221	204	204	197	223	209	181	193	199	195	199
144	151	158	175	162	154	178	192	200	205	204	200	206	194	190	214	216	216	199	195	207	212	188	184	210	191	209

9. CHANNEL REGISTRATION

Registration of digital images of the digitized film processed from a multiple camera array system, has become necessary due to the rotational, translation and scaling errors introduced during processing. A course alignment is made by employing the matching matrix technique but this only corrects for a horizontal or vertical shift in the digital images for alignment. There exist errors in the digital images, such as rotation and translation errors, which have to be corrected by a more effective means. A technique which will accomplish this is presently being developed and has been demonstrated. Computer results are not available, due to contract expiration which prevented finalizing the analysis, therefore, no illustrations are shown.

The technique requires generating boundary maps of the digital images and a manual technique is used to correlate similar areas of the digital images with each other. A good sampling of the entire scene is made which provides enough input to a set of equations (equations are presently being derived and improved upon) to calculate coefficients for each digital image combination. When used with the x and y coordinates of a reference channel, position pointers are calculated giving the respective x and y coordinates of the channels to be aligned. Data points are taken from these coordinates and placed at the coordinates of the reference channel. All channels are then output to magnetic tape becoming merged for further processing.

A program was written to overlay any combination of digital images containing boundary information (0 and 1) that have been registered and merged. This overlay program produces a map showing the location of boundaries and homogeneous areas that are similar. This program is used in conjunction with the registration process and evaluating the accuracy of the registration process.

After the digital images are aligned satisfactorily the registration process can be carried out on the raw data and processed as desired, or the merged boundary tapes can be processed.

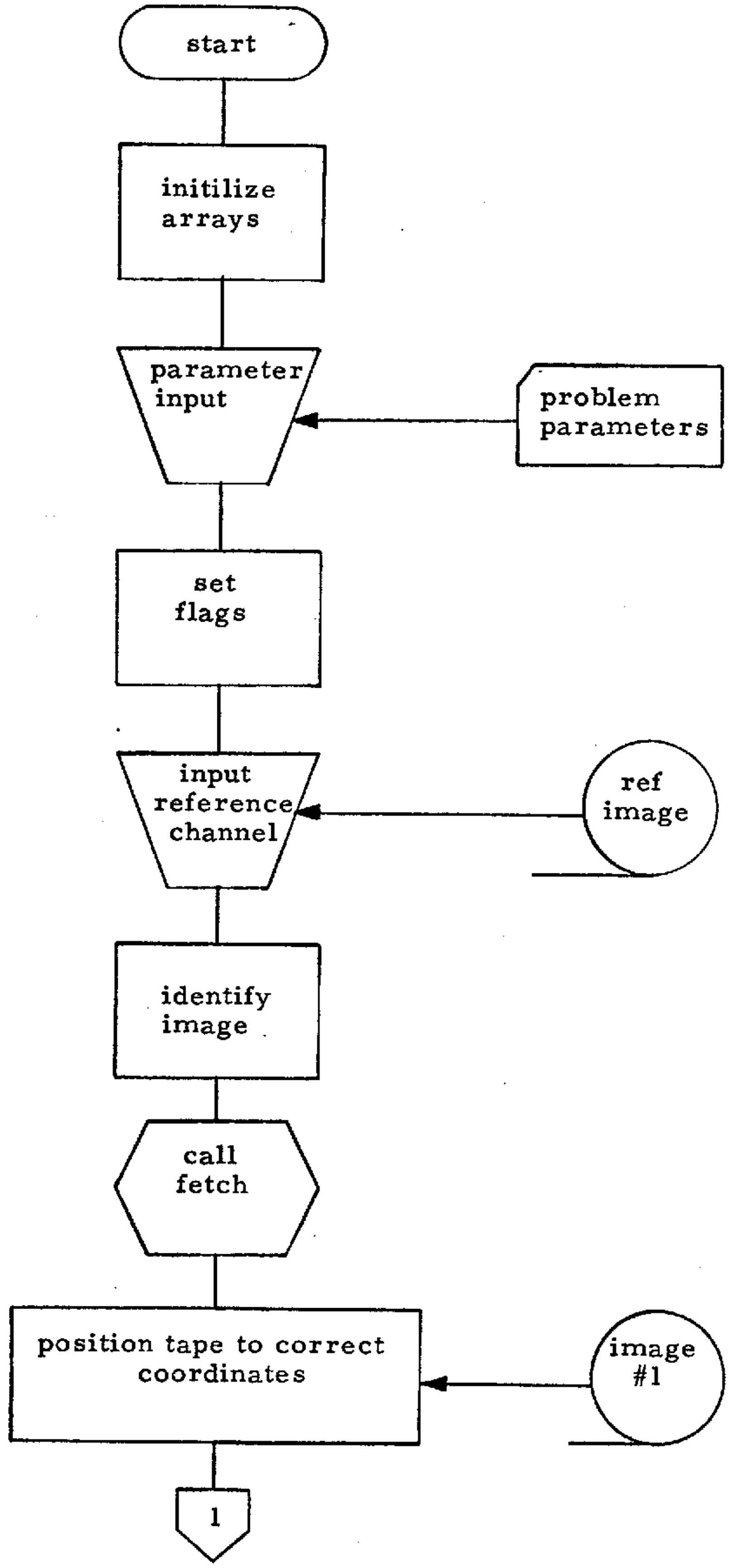
A. Computer Program

A computer program was written to register multiple digital images using boundary information or raw data. The program inputs a reference channel and the x and y coordinates of data points from the reference channel are input to a subroutine FETCH which returns through the call statement the x and y coordinates of the channel to be aligned. Subroutine FETCH contains DATA statements with coefficients used in the calculations. These DATA statements can be changed to reflect coefficients of different digital images.

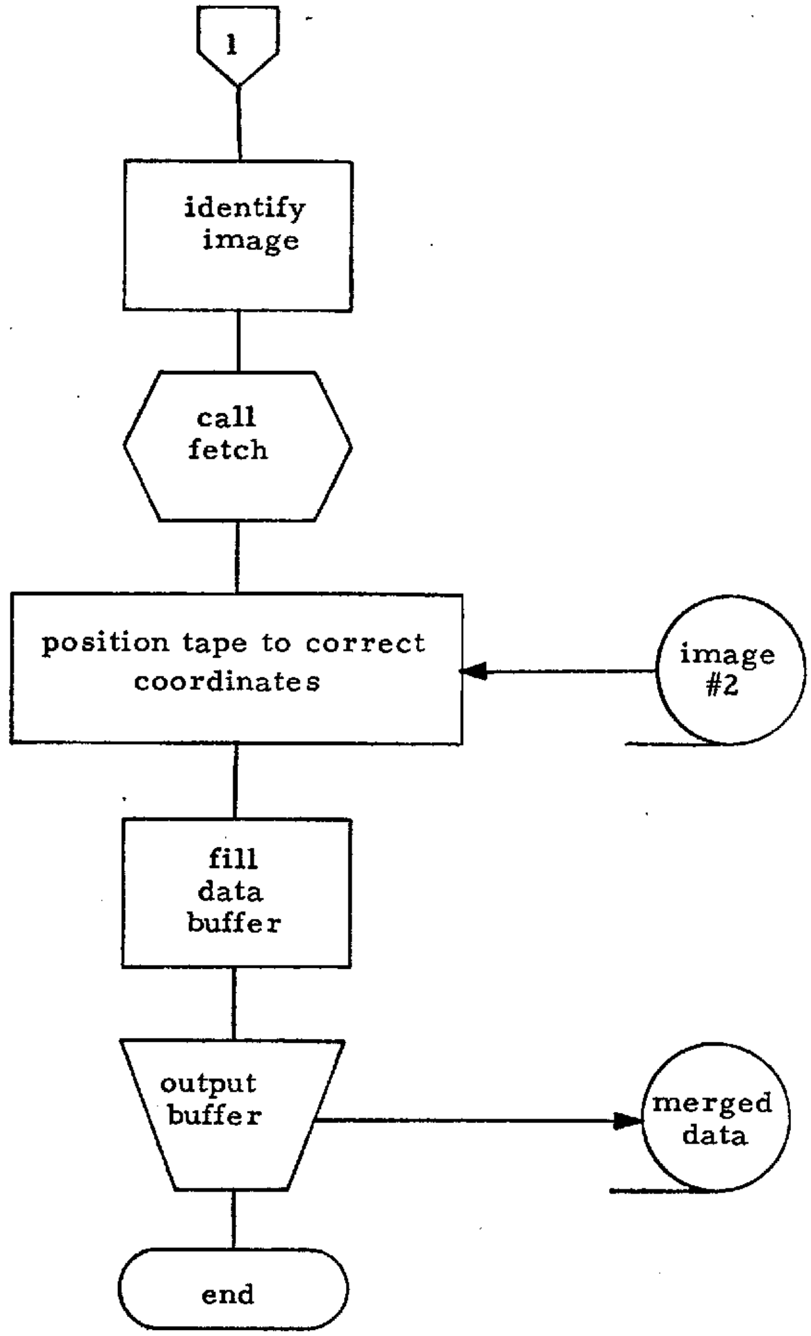
B. Data Problem Parameters

NSPS	Number of samples per scan
NCHAN	Number of channels on input tape
NSCANS	Number of scans to process
NSKIP1	Initial position of tape number one
NSKIP2	Same as above for tape number two
NSKIP3	Same as above for tape number three

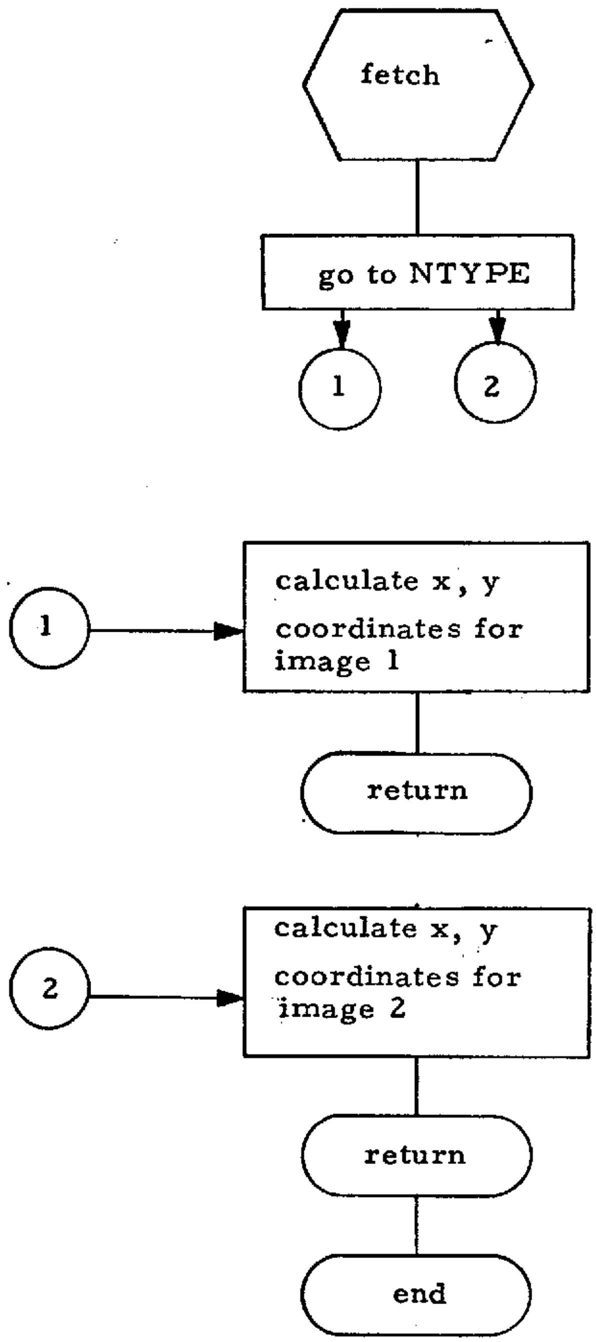
C. Program Flow Chart



Channel Registration



Channel Registration



Channel Registration (Concluded)

10. SPECTRAL DISCRIMINATION

Capabilities have been developed to discriminate ground scene features using computer techniques with digitized images. These capabilities are available at MSFC for analysis and interpretation of earth resources flight data. Computer programs and algorithms were developed during this contract period, March 22, 1971 to September 22, 1972, under the auspices of the National Aeronautics and Space Administration, with Mr. Robert Jayroe COR. The mathematical rationale used in the development of these computer programs are to be published by Mr. Robert Jayroe in a NASA report entitled "Unsupervised Spatial Clustering with Spectral Discrimination."

INTENTIONALLY BLANK

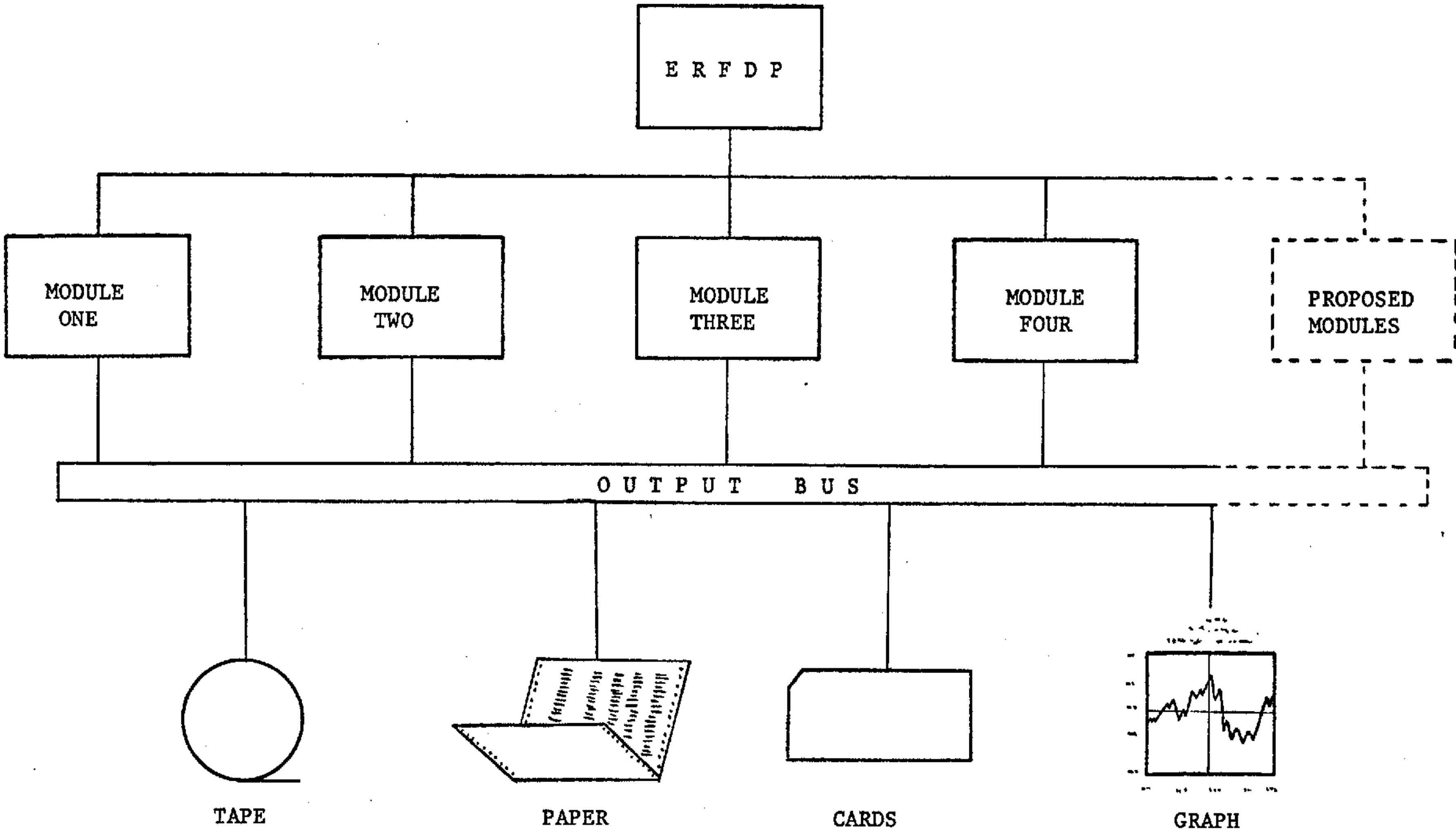
SECTION II INTRODUCTION

Processing earth resources data is the future will require a somewhat automatic approach due to the volume of data and the extent of analysis required to interpret the data. This requires a complex data processor designed to maintain an automatic data processing (ADP) environment for the user. The processor is designed using independent modules with an executive program driving each module. A block diagram depicting this modular structure is shown as Figure 22.

This processor uses an overlay technique, which compiles and communicates with all modules, but only executes modules selected by the user. The concept of using overlays provides capabilities for processing input data, through all the program modules sequentially or only selected modules as desired. Each module can be improved upon, expanded and modified independently of other modules, and each module can be improved or modified in a 32K core storage environment irrelevant to the other modules (see Figure 23).

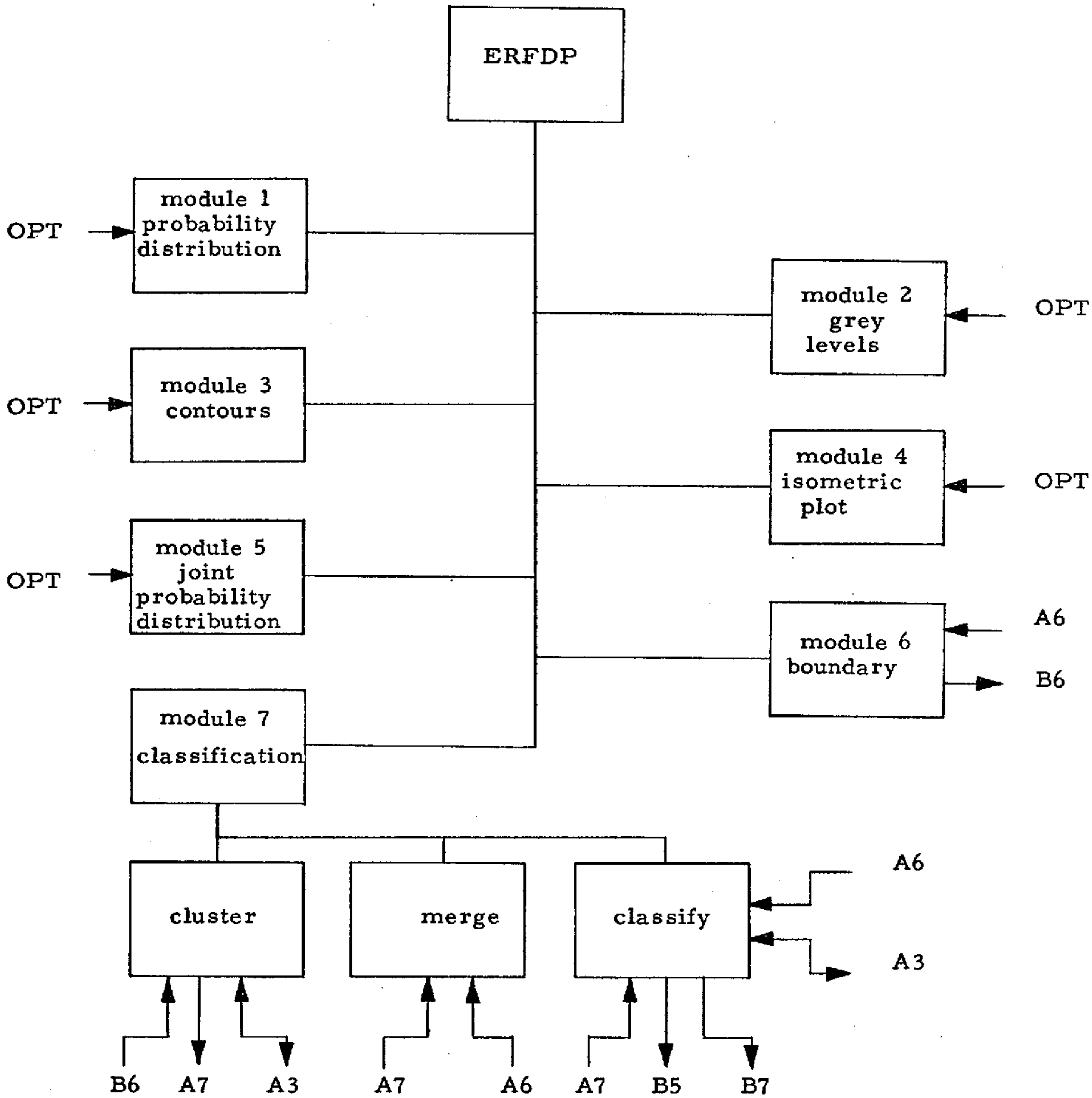
This conept has the disadvantage of having to utilize an intermediate storage tape, which necessitates executing read/write commands, which are certainly time-consuming. However, the advantages outweigh the disadvantages significantly by providing versatility in the overall function. To further improve on the efficiency of the program, the read/write commands use alternate input/output channels when possible to eliminate, in most cases, the computer being in a "wait" condition due to either channel "A" or channel "B" being busy.

The overlay concept entails having a main program which calls each individual module as requested by the user. This keeps the overall processing contained in one large program and each program module is brought into the computer, compiled, overlay structures created, and then output onto an overlay tape. All modules are treated in this fashion regardless of which and how many are to



Earth Resources Flight Data Processing

Figure 22 Block Diagram Depicting Modular Structure



Tape Assignments:
LT1 = A3
LT9 = B5
LT10 = A6
LT11 = B6
LT12 = A7
LT13 = B7
OPT = (Optional)

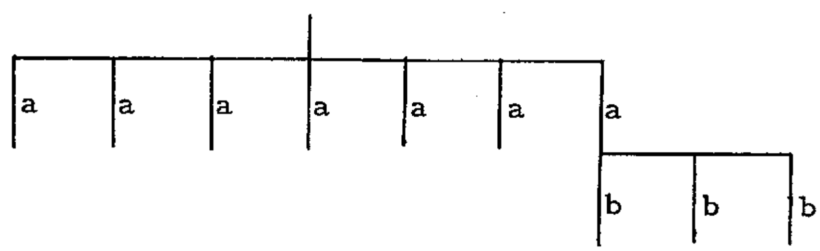
Earth Resources Flight Data Processor

Figure 23 Block Diagram ERFDP

be used. Each module to be executed is brought into the computer in a "transit" area (see Figure 24), and upon completion, another module, if requested, is brought into the same "transit" area to be executed. A common area in the computer is used to store information that may be used by more than one module. Each module has access to this common area at any time it occupies the "transit" area.

1. EARTH RESOURCES FLIGHT DATA PROCESSOR

The ERFDP is comprised of seven separate and independent modules which reside in computer memory only while being executed. Module seven consists of three lower level modules that are called into memory to perform their function automatically when module seven is requested by the user. The segmented structure appears as diagrammed.



Each module can be called into core memory and executed independently and in any order with the exception of module seven. Module six has to be executed prior to module seven, since the boundary mapping output is input to module seven. If module six has been run previously and a boundary mapping output tape is available, then module seven can be executed alone.

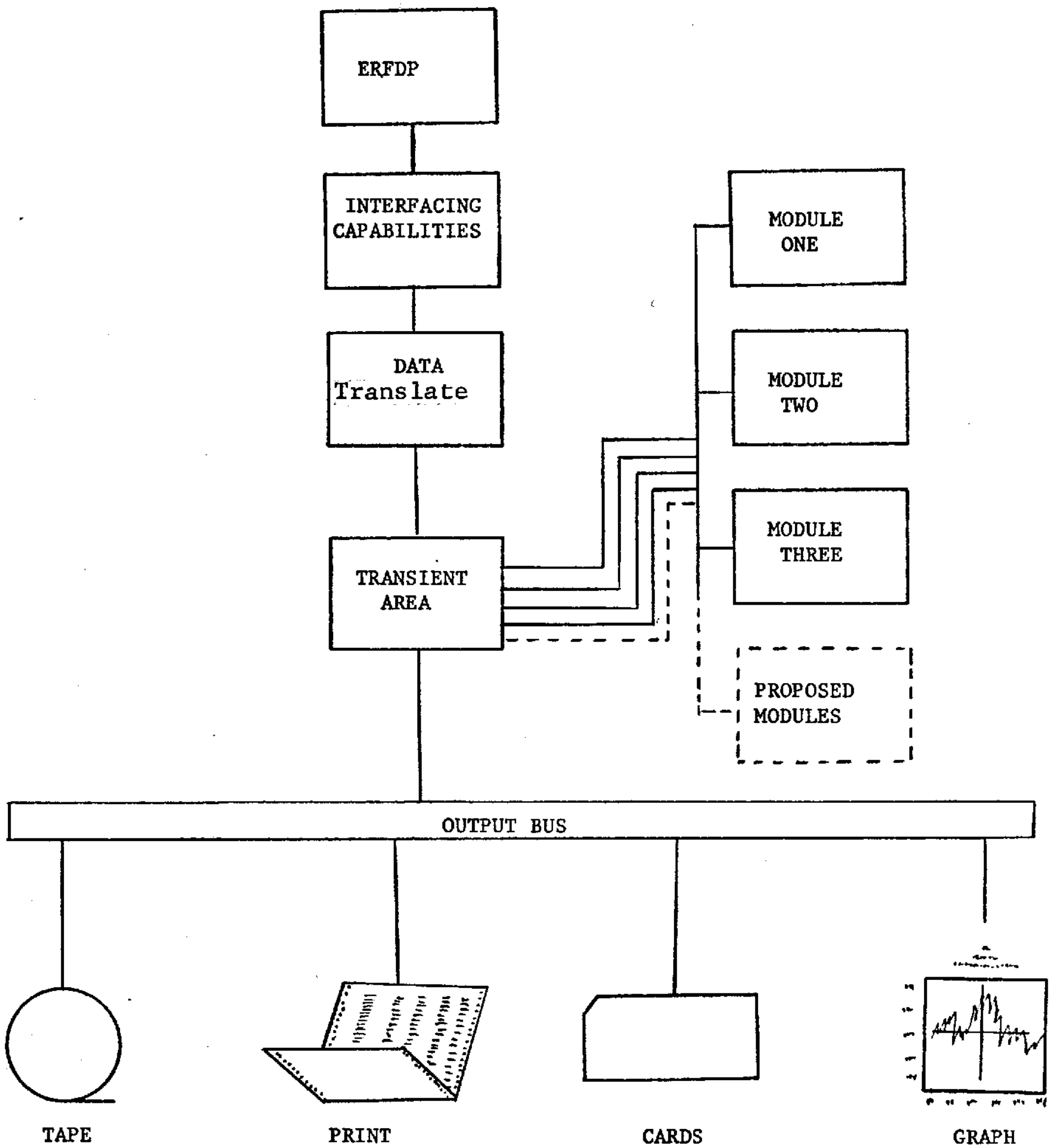


Figure 24 Block Diagram ERFDP

Earth Resources Flight Data Processing

A dimensional variable called MODULE is dimensioned 8, which sets aside eight memory locations to store a users request for module execution. The number of locations filled depends upon the number of modules requested to be executed by the user. The remaining locations contain zeroes and when the processor encounters a zero while examining the list, it terminates under system control.

A. Data Problem Parameters Example Setup

$DATA	
$INIT	
MODULE = 1,2,4,6,7	User request modules 1,2,4,6,7 to be executed and in that order.
$END	

B. Input Tapes

Unit - Users option under individual modules (A6 recommended)
Type - Reference individual modules

C. Output Tapes

Reference individual modules

2. MODULE ONE

Module one when called into memory calculates a probability density function for any number of channels not to exceed 12. The module provides options to calculate probability density functions on selected channels also.

A. Data Problem Parameters Example Setup

$INPUT1	
NCH = 3	Number of input channels
NSPS = 794	Number of samples per record
NSCANS = 500	Number of scans or records to process
NSTART = 1	Starting sample number
NSTOP = 794	Stopping sample number
NBTLG = 12	Bit length of input data word
MODE = 2	Signifying non-FORTRAN input tape
ITYPE = 0	Fixed point input; ITYPE = 1 ; Floating point 36-bit word input
MSFC = 0	Not MSFC scanner format; MSFC = 1; MSFC scanner format
LTN = 10	Logical tape unit of input tape
NSKIP = 2	Skip first two data records
NCRE = 1	Data increment to process
XMAX = 1200.0	Maximum value in data set
XMIN = 300.0	Minimum value in data set
NOCHS = 3	Number of channels to calculate probability density function
NWHICH = 1,2,3	Calculate probability density function on channels 1,2, and 3
$END	
7/8	

B. Input Tapes

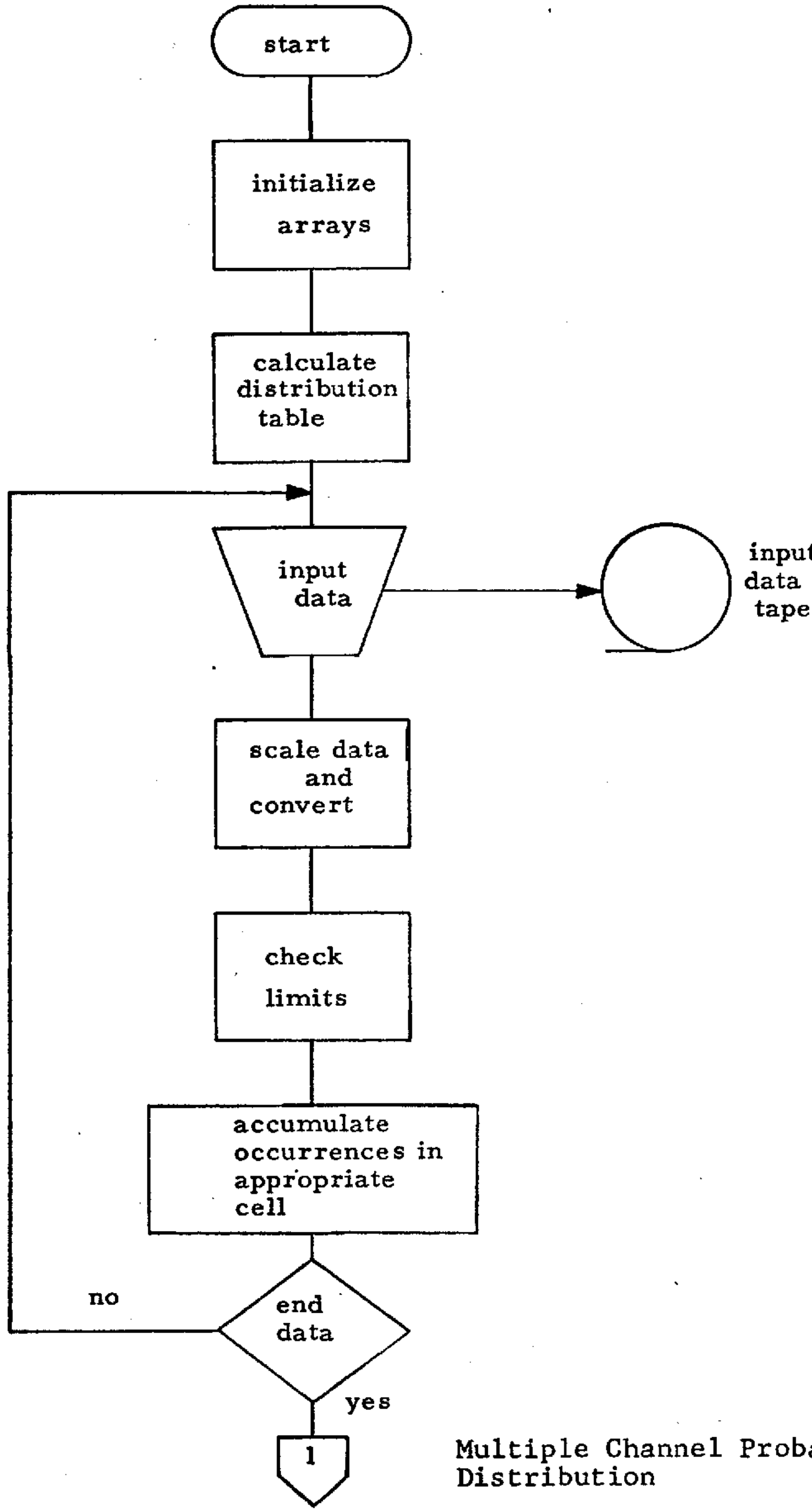
Unit - Users option under input parameters

Type - Any odd parity binary, 3-bit modulus, fixed point with word lengths $\leq$ 36 bits, or floating point.

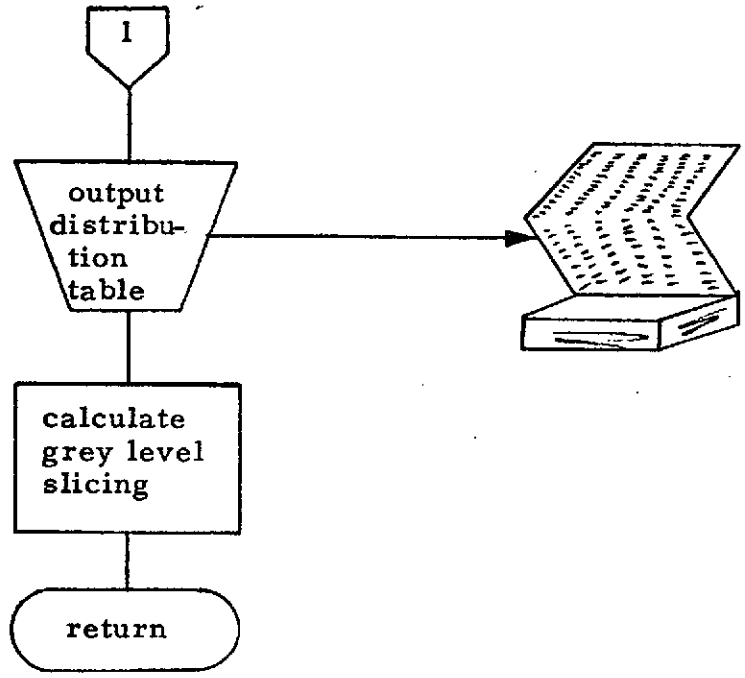
C. Output Tapes

None (only print output)

D. Program Flow Chart



Multiple Channel Probability
Distribution



Multiple Channel Probability Distribution (concluded)

OUTPUT EXAMPLE

PROBABILITY DISTRIBUTION

AMPLITUDE	CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8	CH 9	CH 10	CH 11	CH 12
-1.5	0	0	0	0	0	0	0	0	0	0	0	0
-1.4	0	0	2	0	2	0	0	0	0	0	1	0
-1.3	0	0	3	0	0	0	0	0	0	0	0	0
-1.2	1	0	9	0	5	0	0	1	0	0	1	0
-1.1	0	0	21	0	7	0	0	0	1	1	8	0
-1.0	5	0	29	0	15	1	0	1	0	0	22	0
- .9	4	0	35	0	17	1	0	3	2	4	31	0
- .8	8	1	39	0	34	2	3	9	7	8	47	0
- .7	12	0	48	0	39	8	8	18	20	16	60	0
- .6	24	7	55	1	47	12	12	22	25	30	55	0
- .5	21	11	66	0	51	22	22	38	40	40	40	0
- .4	38	5	58	2	55	31	41	47	48	52	27	0
- .3	45	12	49	12	49	49	70	50	55	60	11	0
- .2	57	22	42	18	44	66	82	54	62	72	6	5
- .1	59	37	33	20	39	72	96	47	51	55	5	1
0	62	44	25	35	33	62	79	41	45	52	0	8
.1	58	48	19	30	25	40	62	35	39	31	1	16
.2	51	49	12	41	18	22	38	25	25	20	0	27
.3	47	56	6	54	7	12	12	17	16	15	0	32
.4	35	52	1	61	8	8	10	12	8	7	0	39
.5	25	40	0	55	2	2	7	8	5	3	0	47
.6	22	32	1	45	1	0	2	0	1	5	0	52
.7	8	24	0	40	0	1	1	2	0	1	0	65
.8	4	16	0	35	1	0	0	0	0	0	0	72
.9	[illegible]	8	0	28	0	0	0	0	0	0	0	58
1.0	3	7	0	20	0	0	0	0	0	0	0	46
1.1	1	1	0	12	0	0	0	0	0	0	0	39
1.2	0	0	0	7	0	0	0	0	0	0	0	22
1.3	1	1	0	0	0	0	0	0	0	0	0	7
1.4	0	0	0	1	0	0	0	0	0	0	0	1
1.5	0	0	0	0	0	0	0	0	0	0	0	0

NUMBER OF OCCURRENCES

3. MODULE TWO

Module two outputs on print paper and/or Stromberg-Carlson 4020 recorder, the quantized levels of a data set with alphanumeric characters displaying specific quantized levels. Alphanumeric characters can be selected and input to the module on a card included in the input problem parameters to provide different shading for certain levels. This module displays only one channel per computer pass, and that one channel is optional and selected by the user.

A. Data Problem Parameters Example Setup

$INPUT2

NCH = 3	Number of channels on input tape
NSPS = 794	Number of samples per record or scan
NSCANS = 500	Number of scans to process
NSKIP = 2	Number of scans to skip before processing
NSTART = 1	Starting sample number
NSTOP = 120	Stopping sample number
ITERM = 6	Number of passes processing 120 samples each pass
N = 11	Number of levels plus 1
ICHAN = 3	Channel used in mapping
IPRT = 1	Option to print
IPLT = 1	Option to plot
INCX = 0	Increment in x direction for each sample (rasters)
INCY = 8	Increment in y direction for each sample (rasters)
NSTX = 0	Starting x coordinates on plot frame
NSTY = 0	Starting y coordinates on plot frame
NBTLG = 12	Bit length of input data words
MODE = 2	Signifies non-FORTRAN
ITYPE = 0	Fixed point input data; ITYPE = 1 ; Input data floating point 36-bit words
MSFC = 0	Not MSFC scanner format; MSFC = 1; MSFC scanner format

```
NCRE = 1              Data increment
LTN = 10              Logical tape unit of input tape
IOPT = 0              Signifies automatic quantization
$END
$GRYLYL
GLVL = 1., 2., 3., 4., 5., 6., 7., 8., 9., 10., 11.
       1., 2., 3., 4., 5., 6., 7., 8., 9., 10., 11.
           o
           o
           o
       1., 2., 3., 4., 5., 6., 7., 8., 9., 10., 11.
$END
7/8
```

B. Input Tapes

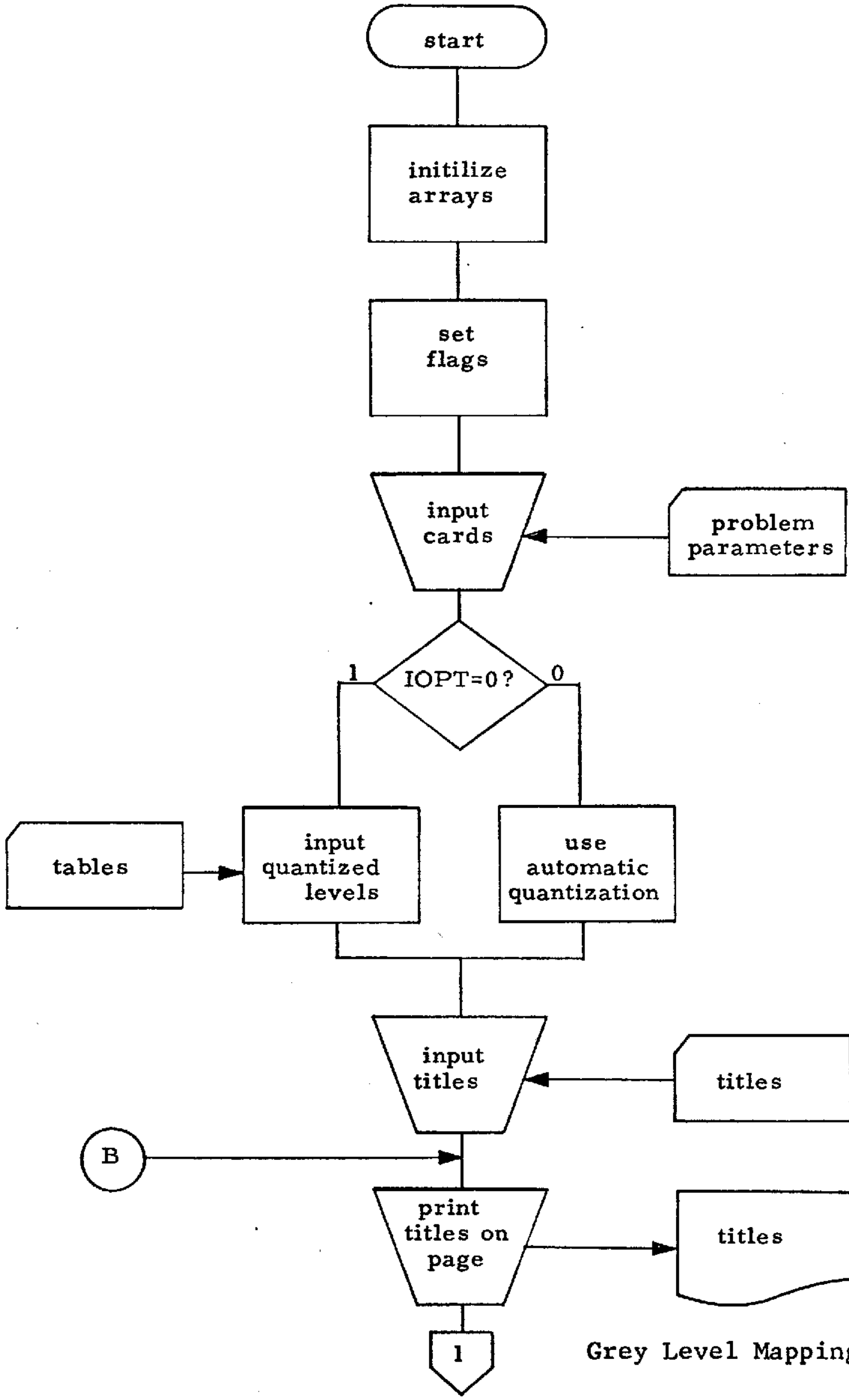
Unit - Users option under input parameters

Type - Any odd parity binary, 3-bit modulus, fixed point, with word length $\leq$ 36 bits or floating point.

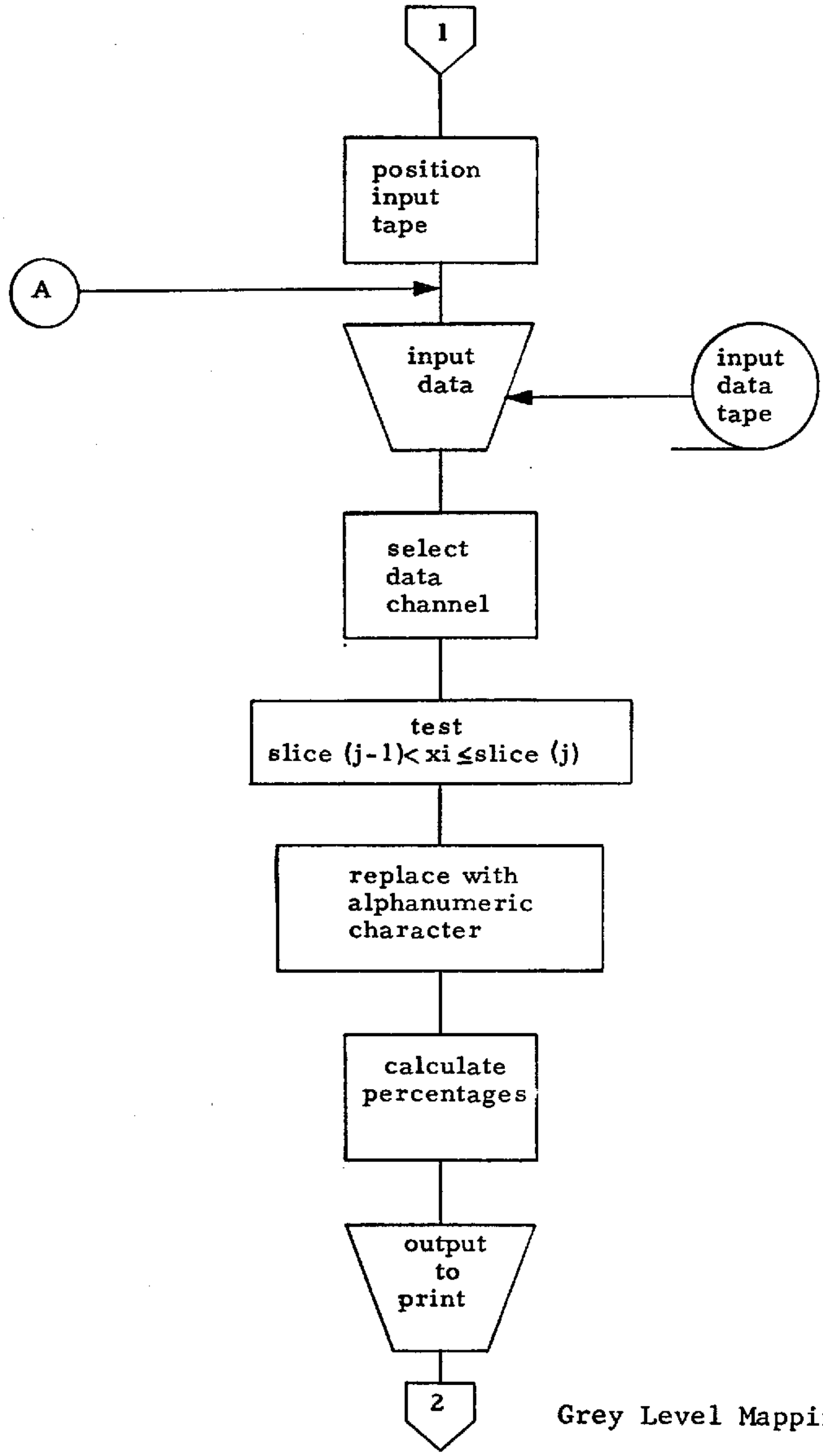
C. Output Tapes

None (only print output)

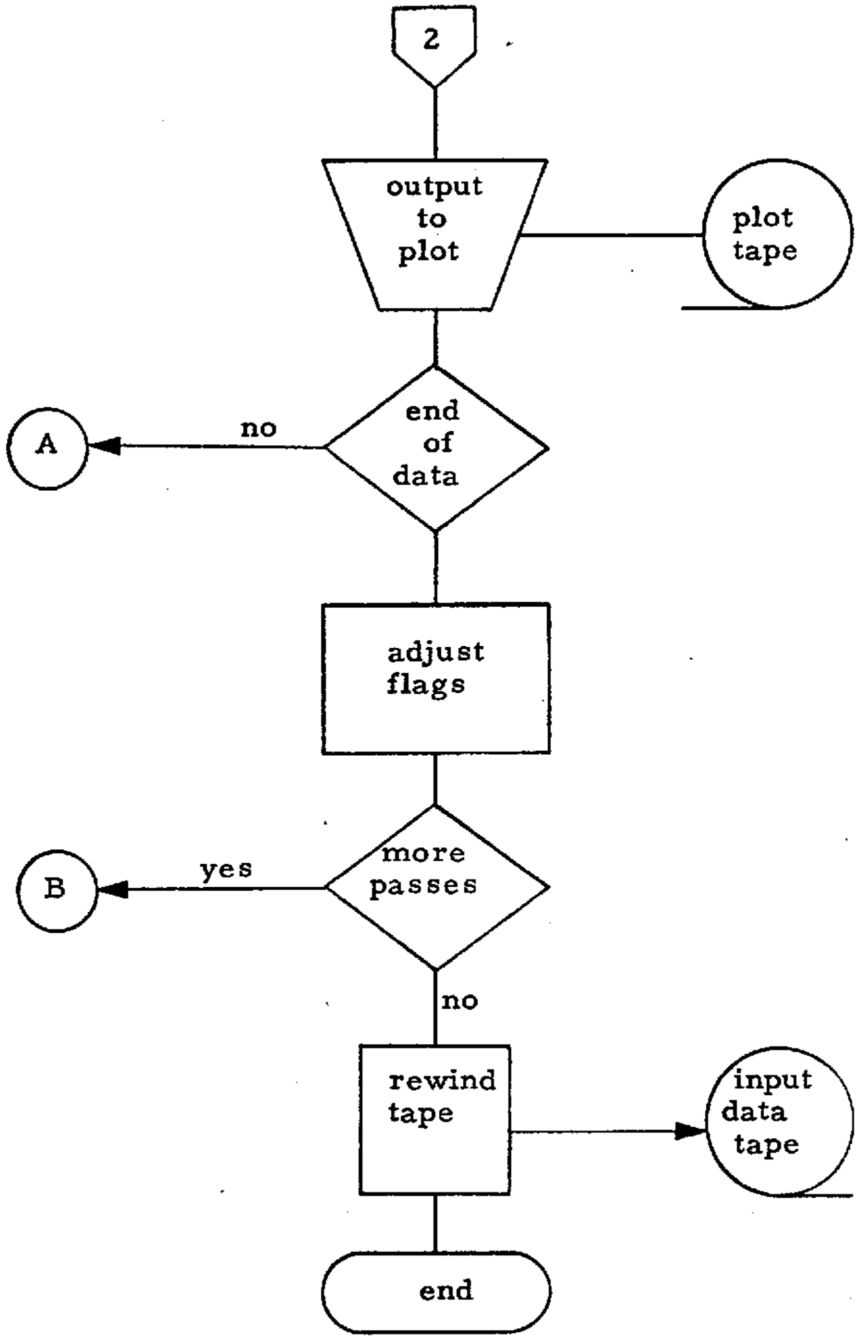
D. Program Flow Chart



Grey Level Mapping



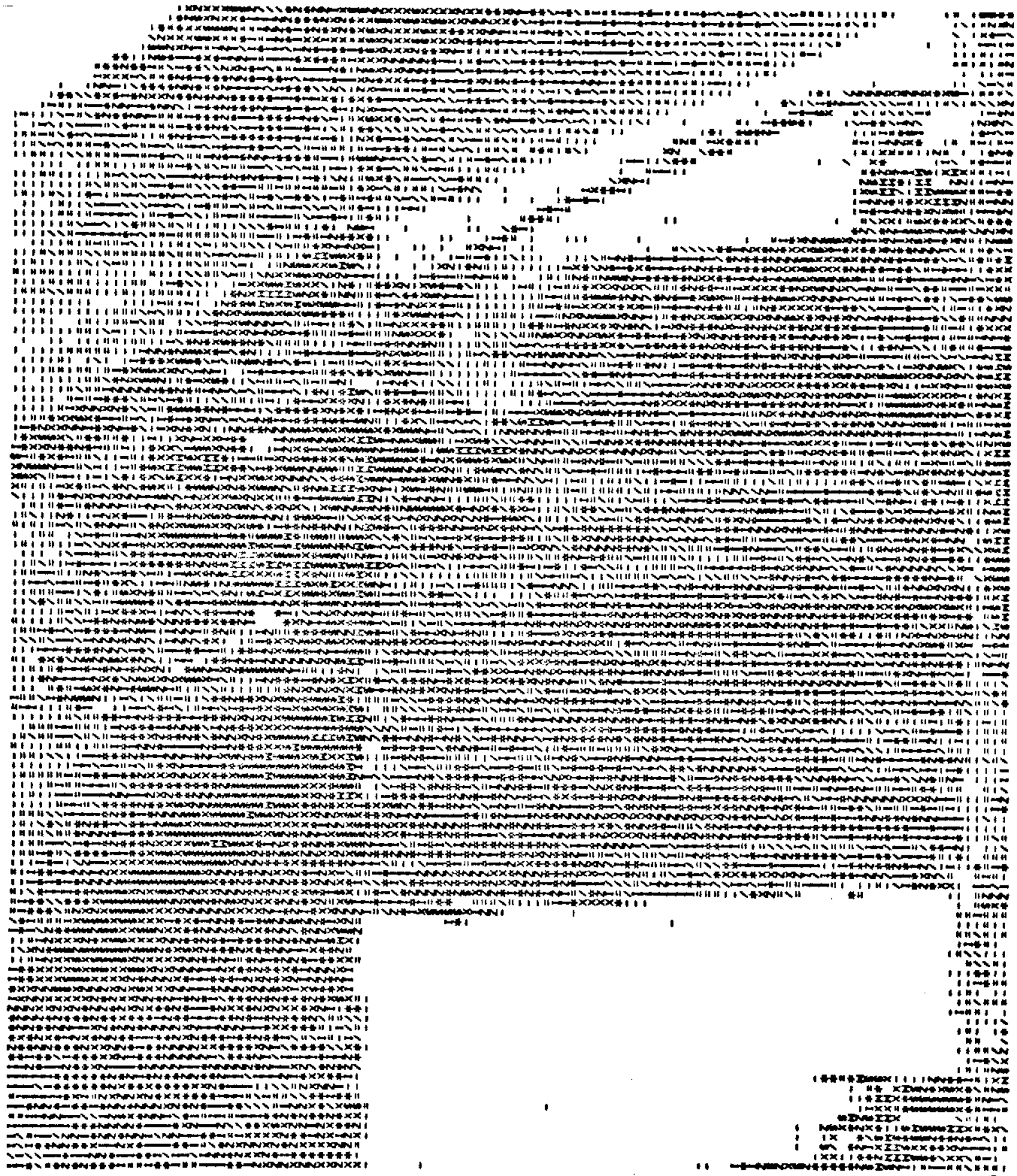
Grey Level Mapping



Grey Level Mapping
(concluded)

OUTPUT EXAMPLE
PURDUE FLIGHT LINE C1
AUTOMATIC QUANTIZED GREY LEVEL

RESOLUTION ELEMENT 1-111   SCANS 1-120



SCAN

RESOLUTION ELEMENT

OUTPUT EXAMPLE
PURDUE FLIGHT LINE C1
AUTOMATIC QUANTIZED GREY LEVEL

RESOLUTION ELEMENT 112-222   SCANS 1-120

SCAN

RESOLUTION ELEMENT

4. MODULE THREE

Module three calculates contour lines throughout an input data set depicting elevation, altitude, temperature or boundary contours. The contour lines are output and displayed by the Stromberg-Carlson 4020 plotter. Frame butting in the module pr vides for continuous contour plotting for infinite data sets. The contour lines can be scaled by utilizing the block size option (BLK) in the input problem parameters.

A. Data Problem Parameters

$INPUT3	
NCH = 1	Number of channels
NSPS = 90	Number of samples in a scan (record)
IRW = 11	Logical tape unit for input data
NCHAN = 1	Channel number
NSNCRE = 1	Number of scans to increment
NPCRE = 1	Number of samples to increment
NPTSL = 1	Lower starting point
NPTSU = 90	Upper stopping point
NBTLG = 36	Bit length of input data word
MODE = 1	FORTRAN formated (1) or non-FORTRAN (2)
ITYPE = 0	Fixed point input ITYPE = 1, floating point input
MSFC = 0	MSFC = 1; MSFC scanner format
NSKIP = 0	Number of records to skip
MAXSCN = 100	Total number of scans to process
MSZX = 90	Data block size in x (samples)
MSZY = 8	Data block size in y (scans)
BLK = 8.0	Plot frame block size in rasters
FHINC = 10.0	Labeling increment
ZMIN = 8.0	Minimum label
ZMAX = 138.0	Maximum label
LAB = 1	Label every interval
$END	
7/8	

B. Input Tapes

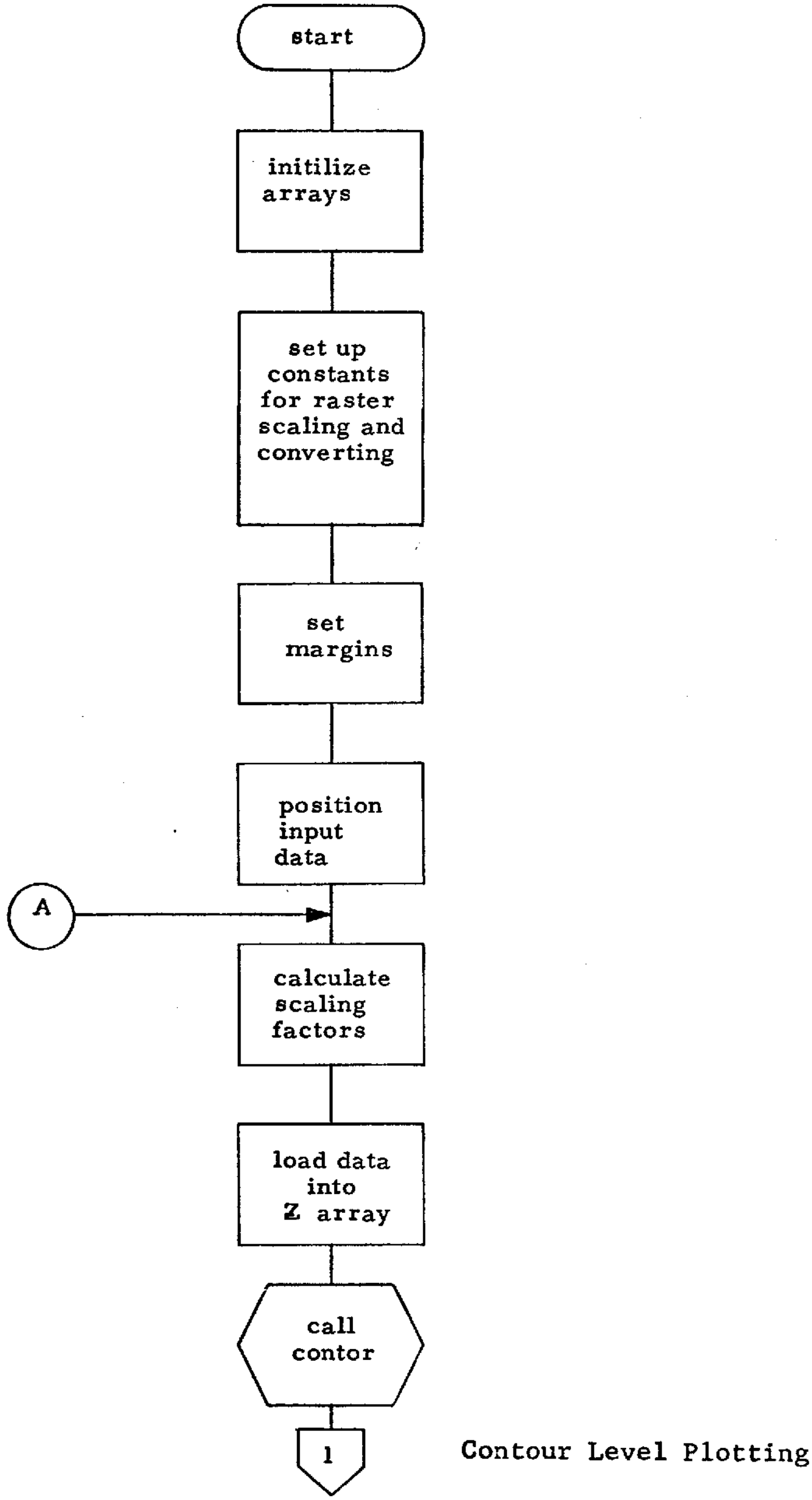
Units - Optional under input data

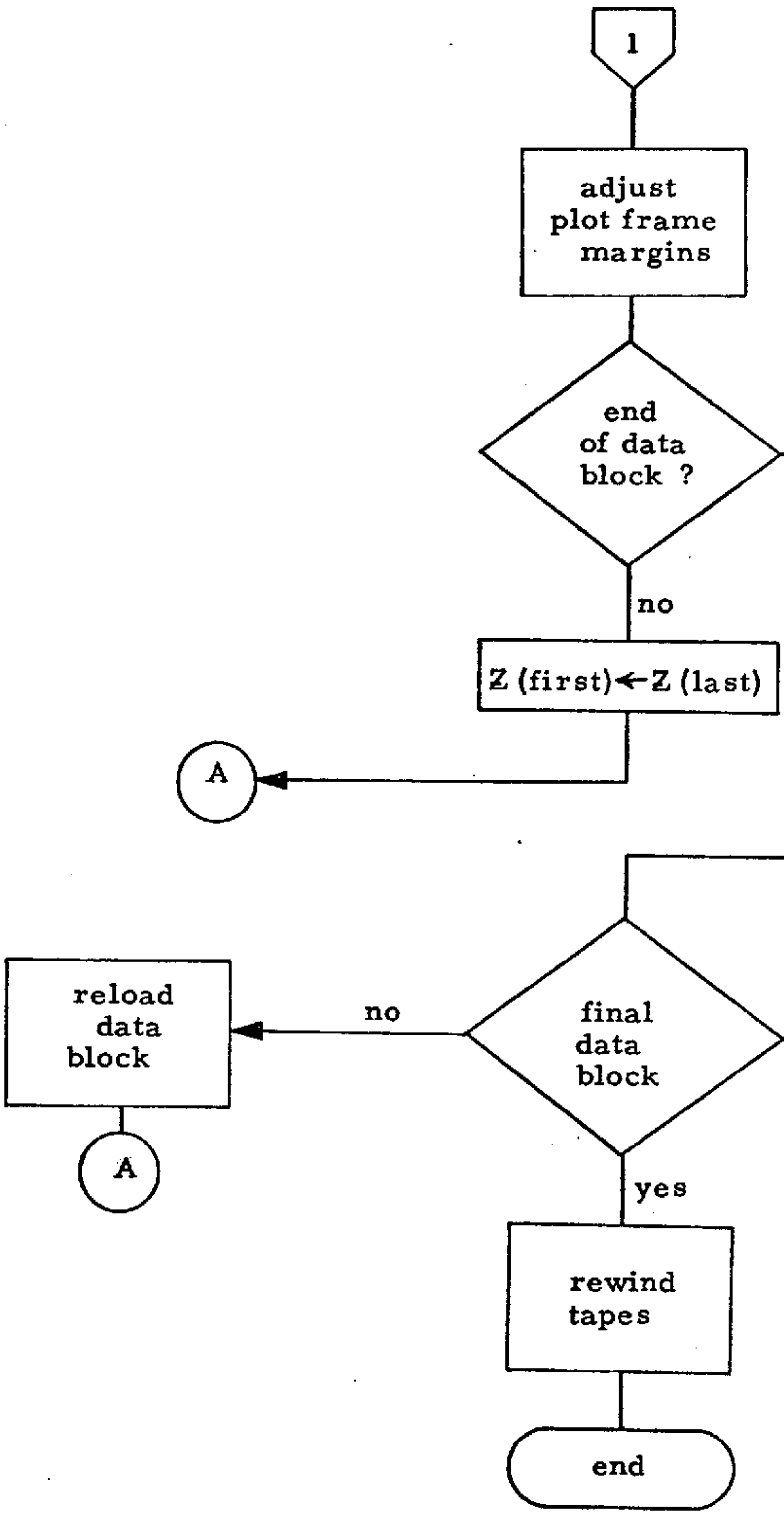
Type - Any odd parity binary, 3-bit modulus, fixed point, with word lengths $\leq$ 36 bits, or floating point.

C. Output Tapes

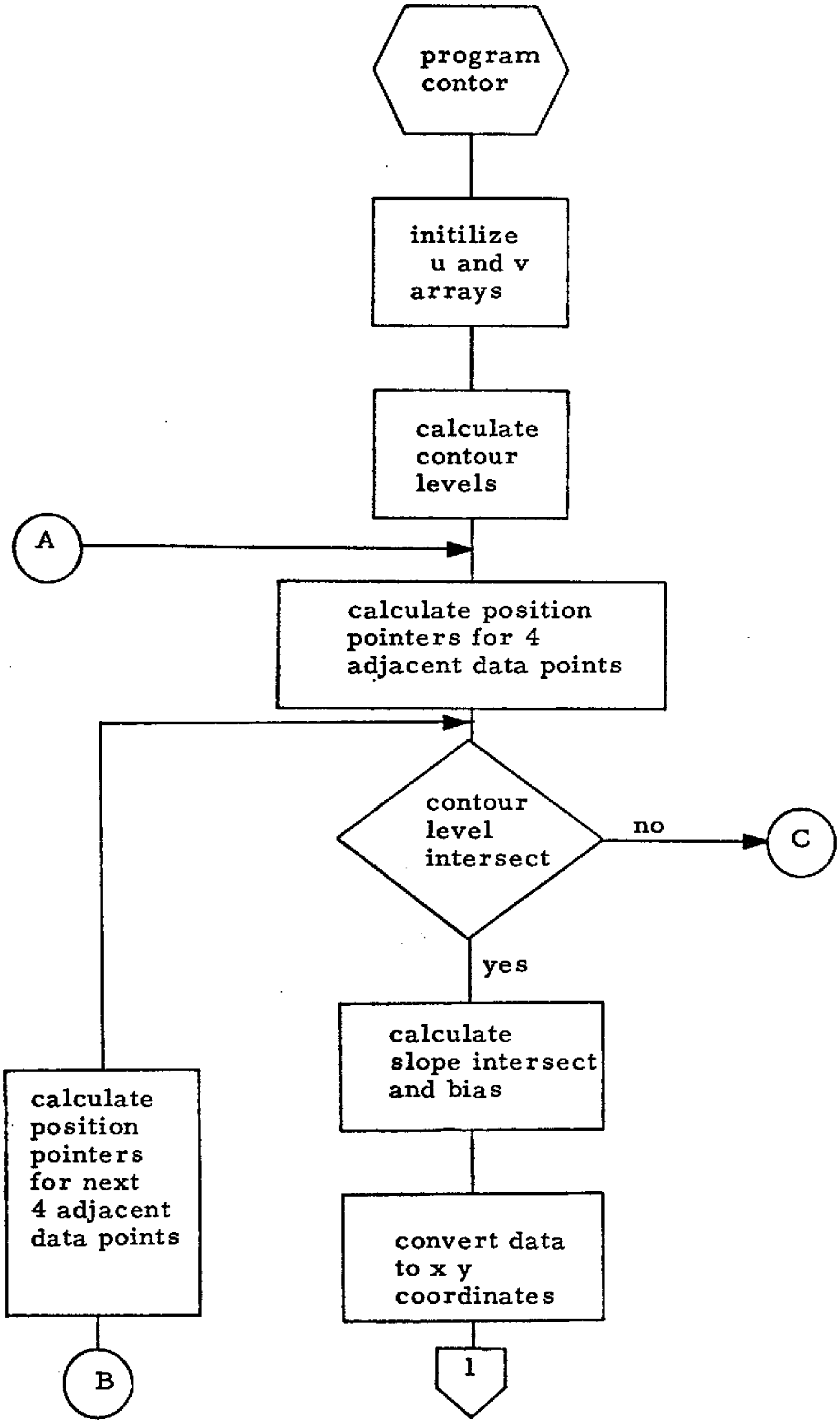
SC 4020 Stromberg-Carlson formatted tape

D. Program Flow Chart

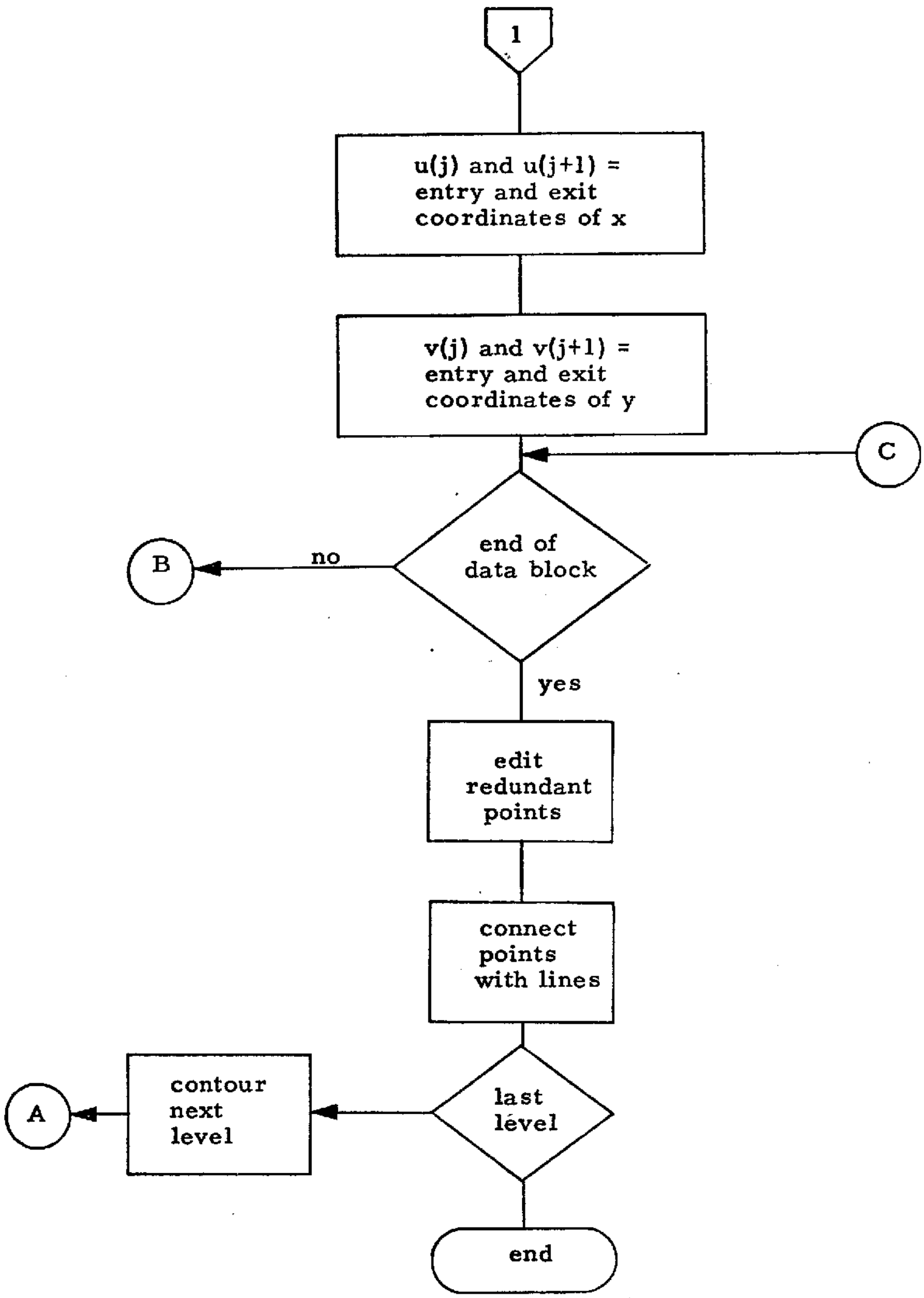




Contour Level Plotting



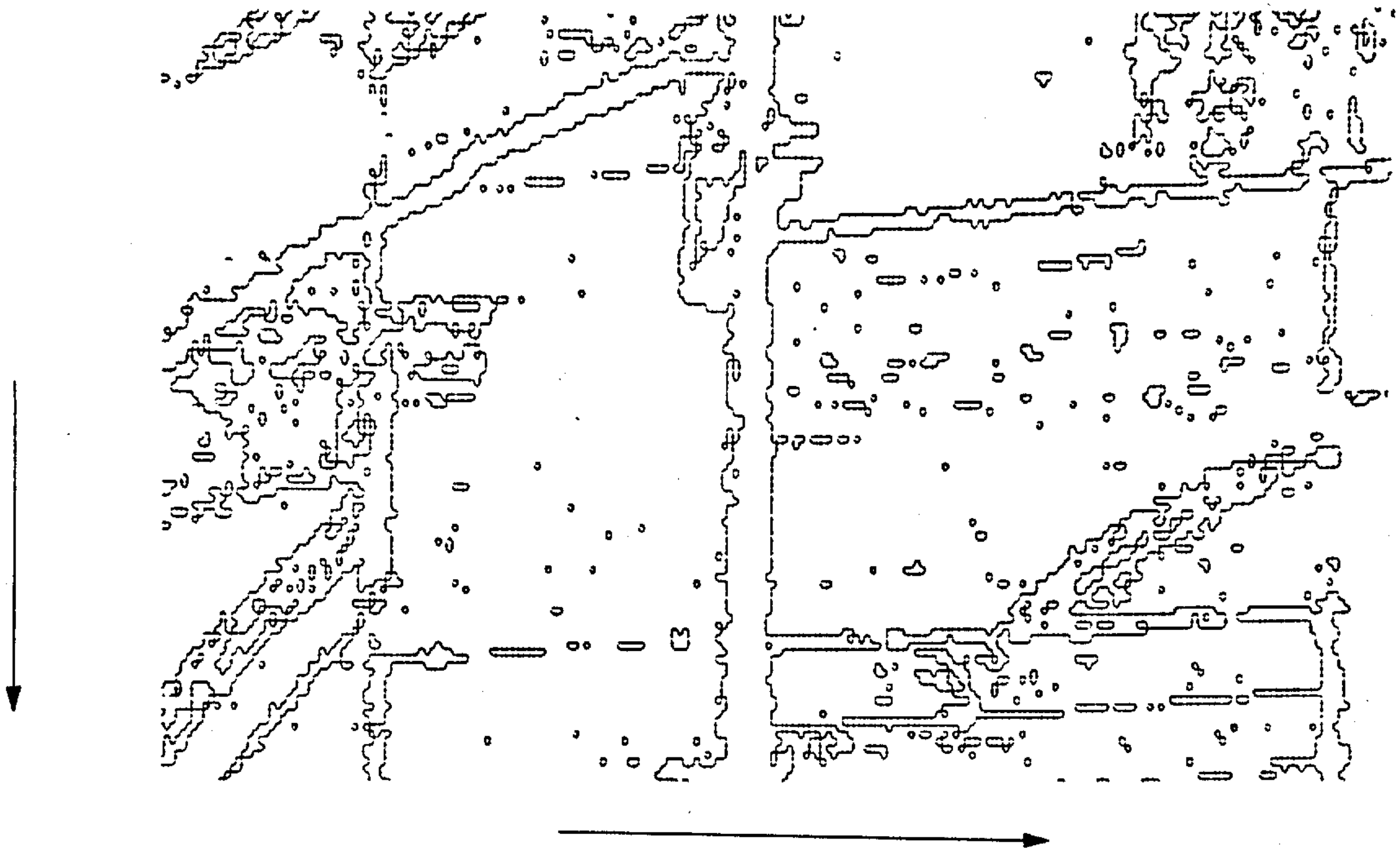
Contour Level Plotting



Contour Level Plotting
(concluded)

OUTPUT EXAMPLE

SCAN 1-120 RESOLUTION ELEMENT 1-222



SCAN

RESOLUTION ELEMENTS

BOUNDARY CONTOURS (PURDUE C1)

5. MODULE FOUR

Module four displays a two-dimensional plot of three-dimensional data where only one single array of data is input and the data samples are ordered in an XY coordinate system, where Y is the scan line count and X is the resolution element in the scan. Data sets of infinite length can be displayed on the SC4020 recorder. The module provides for multiple passes through the data set in order to view the entire data set in sections.

A. Data Problem Parameters Example Input

$INPUT4	
NCH = 12	Number of channels on input data tape
NSPS = 222	Number of resolution elements per scan
NSKIP = 0	Initial records to skip
NBTLG = 12	Bit length of the input data word
MODE = 1	FORTRAN formatted (1) or non-FORTRAN formatted
IRW = 10	Logical tape unit to load input
NCHAN = 6	Channel selected to be plotted
NSNCRE = 1	Scan incrementation
NPCRE = 1	Resolution element incrementation
ITYPE = 1	Fixed point input (0) or Floating point input (1)
MSFC = 0	MSFC scanner format (1) or not MSFC scanner format
NPTSL = 1	Lower resolution element to start
NPTSU = 111	Upper resolution element to stop
MAXSCN = 120	Scan lines to be processed
XMIN = 0.0	Minimum value for scaling the X axis
XMAX = 222.0	Maximum value for scaling the X axis
YMIN = 0.0	Minimum value for scaling the Y axis
YMAX = 255.0	Maximum value for scaling the Y axis
NBLSZX	The distance separating the plotted points in the X direction

NBLSZY	The distance separating the plotted points in the Y direction
NSECT = 2	Two passes through the data to plot the data set in two sections
NSMOV = 0	No smoothing on input data
NDIREC = 1	Orientation of the isometric left view (-1) or right view (1)

$END

7/8

B. Input Tapes

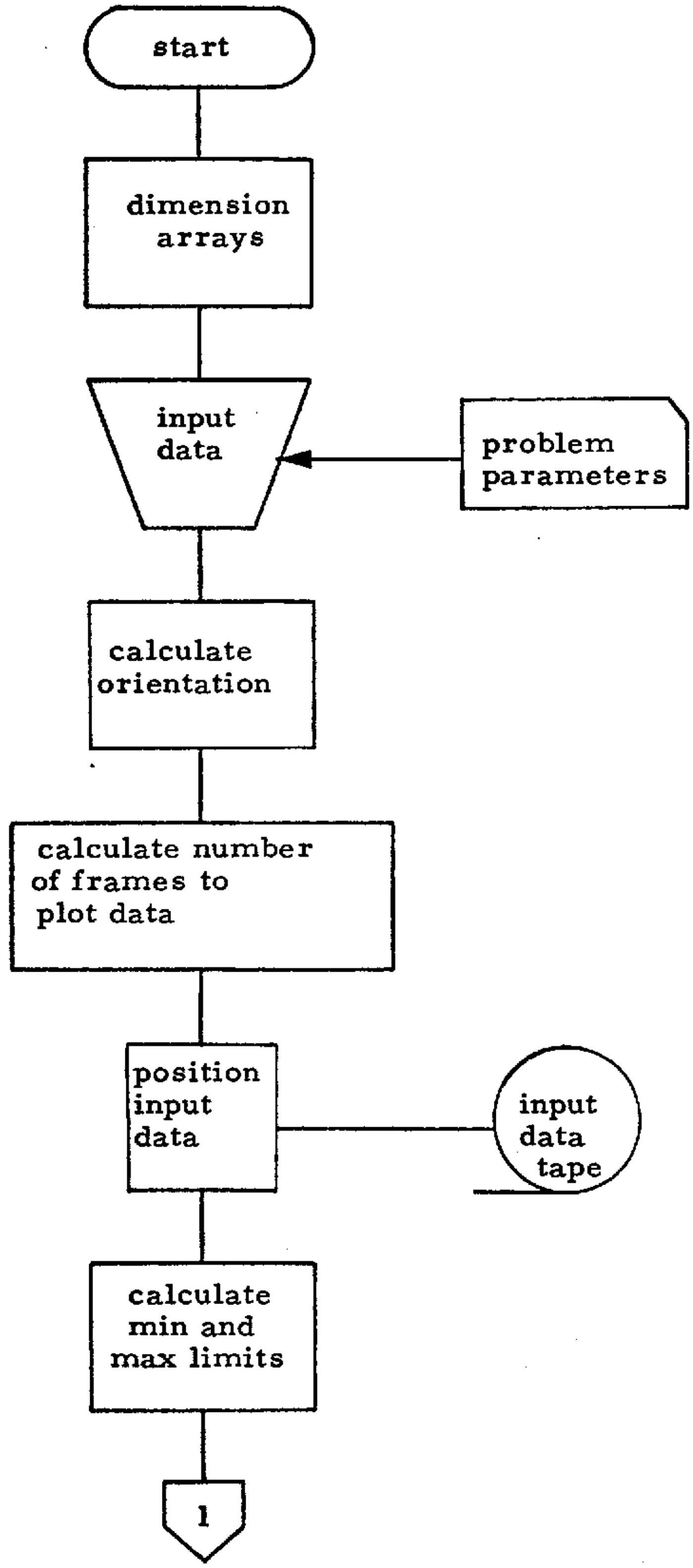
Unit - Users option under input parameters

Type - Any odd parity binary, 3-bit modulus, fixed point with word lengths $\leq$ 36 bits, or floating point binary.

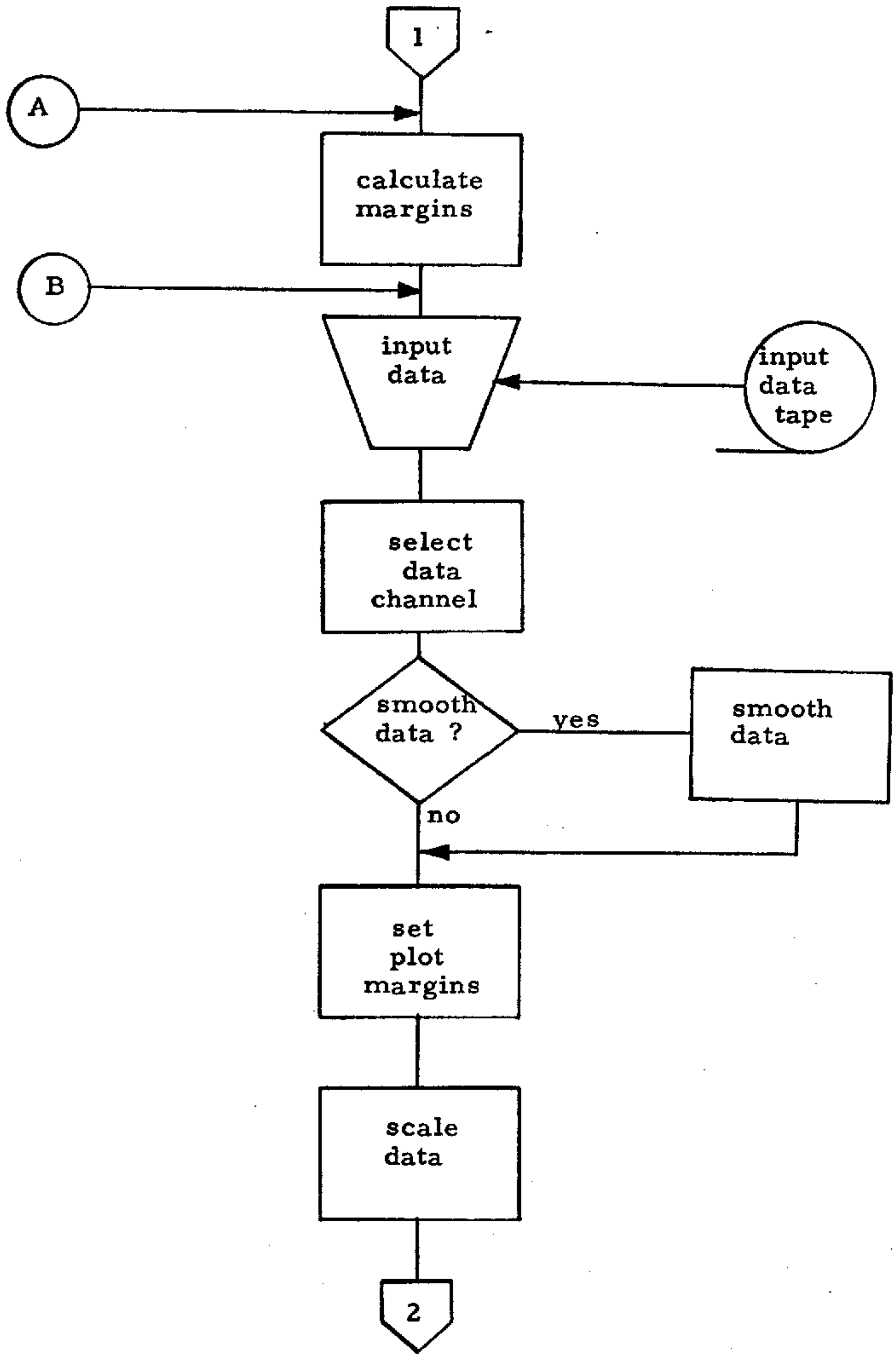
C. Output Tapes

Unit - A8 SC4020

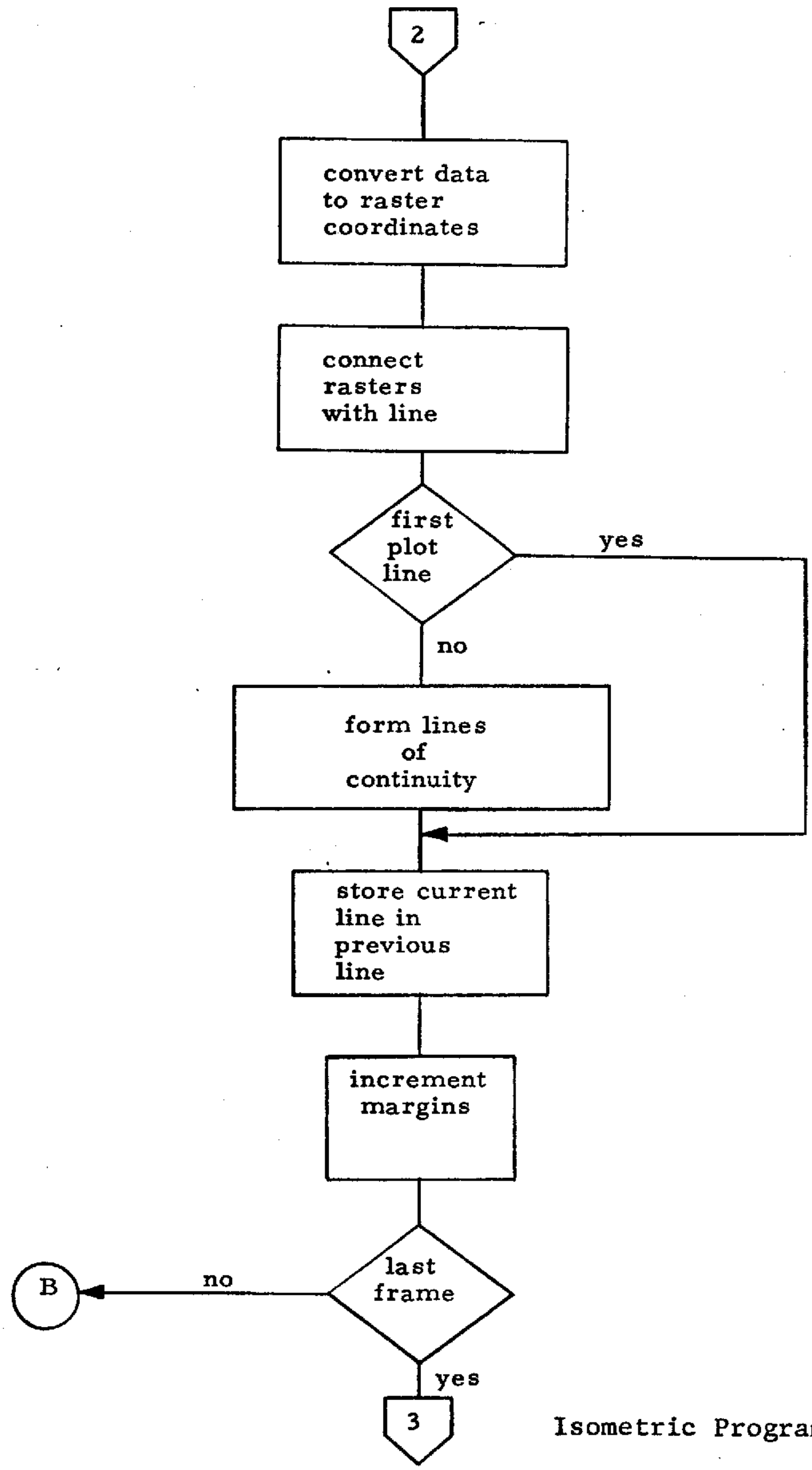
D. Program Flow Chart



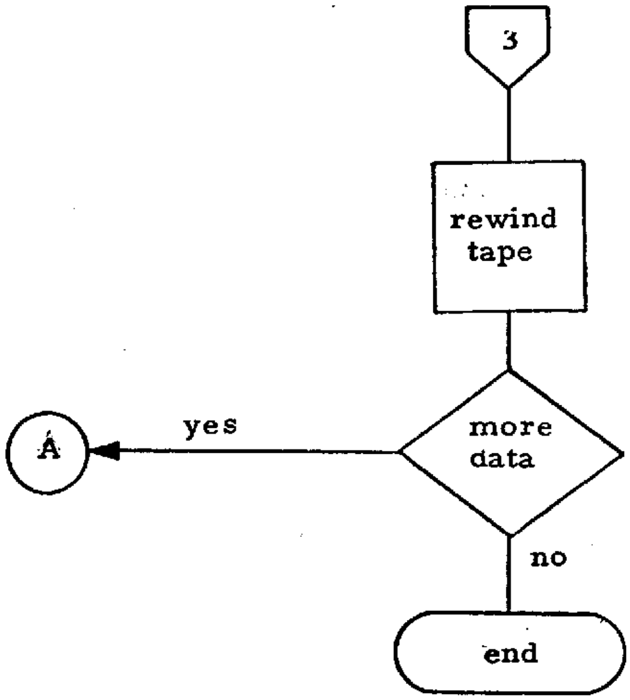
Isometric Program Display



Isometric Program Display



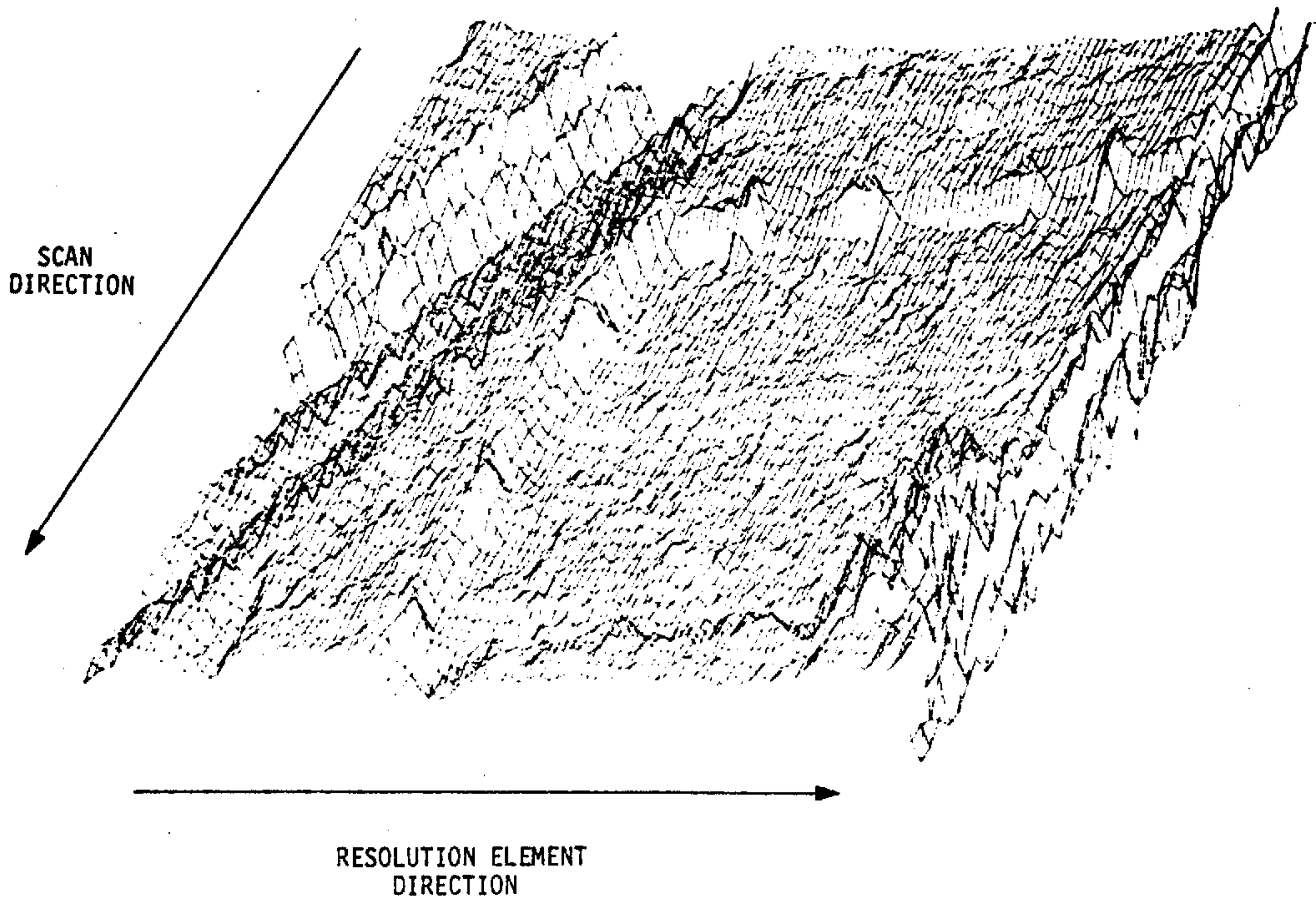
Isometric Program Display



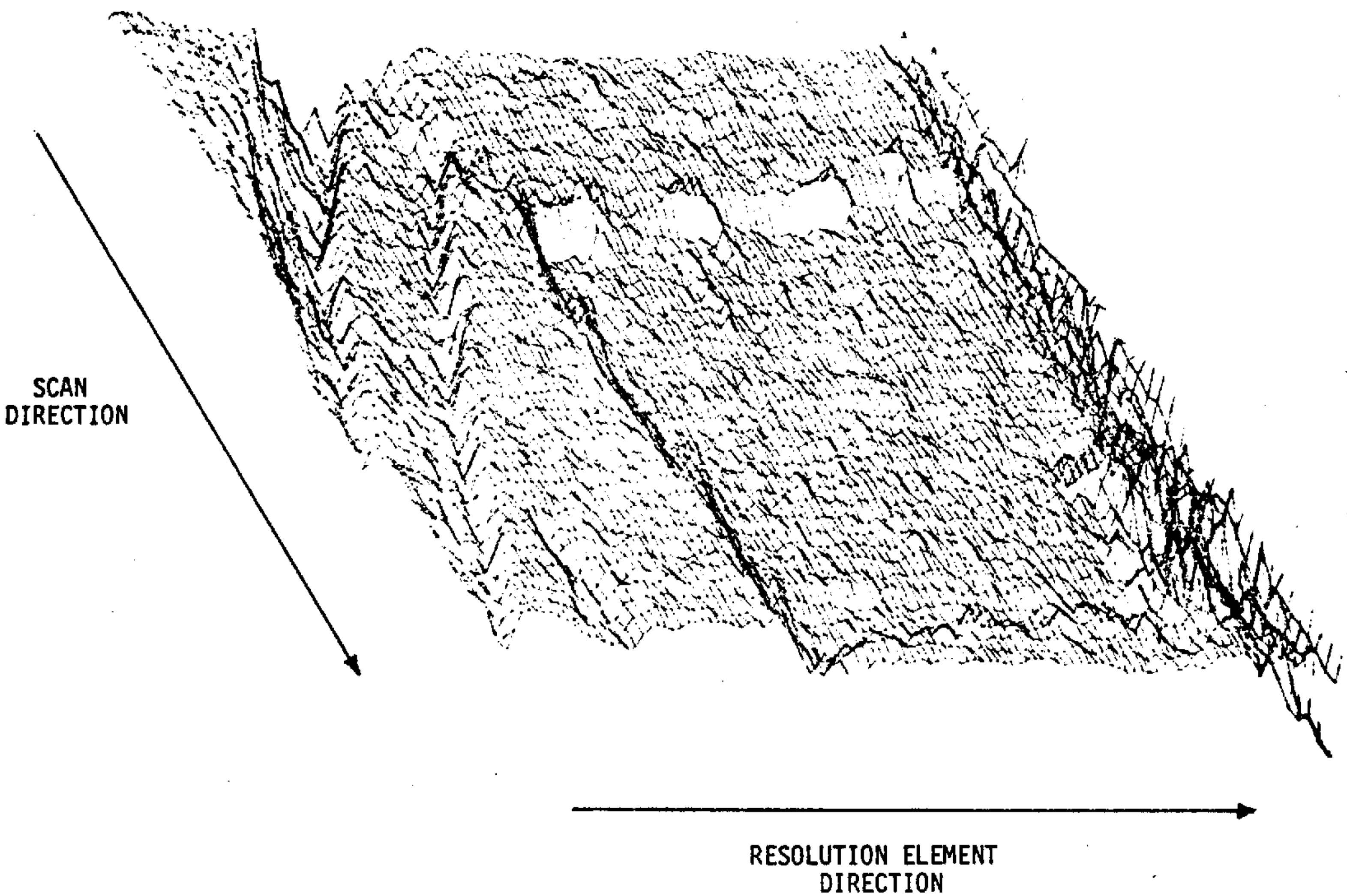
Isometric Program Display
(concluded)

OUTPUT EXAMPLE

FLIGHT LINE C1
SCANS 1-43
RESOLUTION ELEMENTS 1-111
ROTATED CLOCKWISE
CHANNEL 8 0.58-0.62 (MICRONS)



OUTPUT EXAMPLE
FLIGHT LINE C1
SCANS 1-43
RESOLUTION ELEMENT 1-111
ROTATED COUNTER-CLOCKWISE
CHANNEL 8 0.58-0.62 (MICRONS)



6. MODULE FIVE

Module five calculates a joint probability density function from two time-series data traces selected by the user in the data problem parameters. The program provides for multiple passes through the data set for multiple joint probability density functions. The raised data points are classified and stored in one single array along with its associated occurrences of like data pairs. The joint pairs are sorted vertically in descending order and horizontally for each print line in ascending order.

A. Data Problem Parameters Example Setup

$INPUT5

NCH = 12	Total number of channels on input tape
NSPS = 222	Samples per logical record or scan line
NSCANS = 920	Number of logical records or scan lines to process
NSKIP = 0	Initial physical records to skip before processing
NSTART = 2	Starting sample number
NSTOP = 222	Stopping sample number
LTN = 10	Logical FORTRAN IV tape unit
NOJP = 2	Number of paired joint probabilities to process
IMX = 4,3	Use channel 4 and 3 in X
IMY = 6,7	Use channel 6 and 7 in Y
SCALE = 1.0	Used to scale 1
BIAS = 0	Used to shift data

$END

1. 1.),/* ABC... (Alphanumeric characters)

7/8

B. Input Tapes

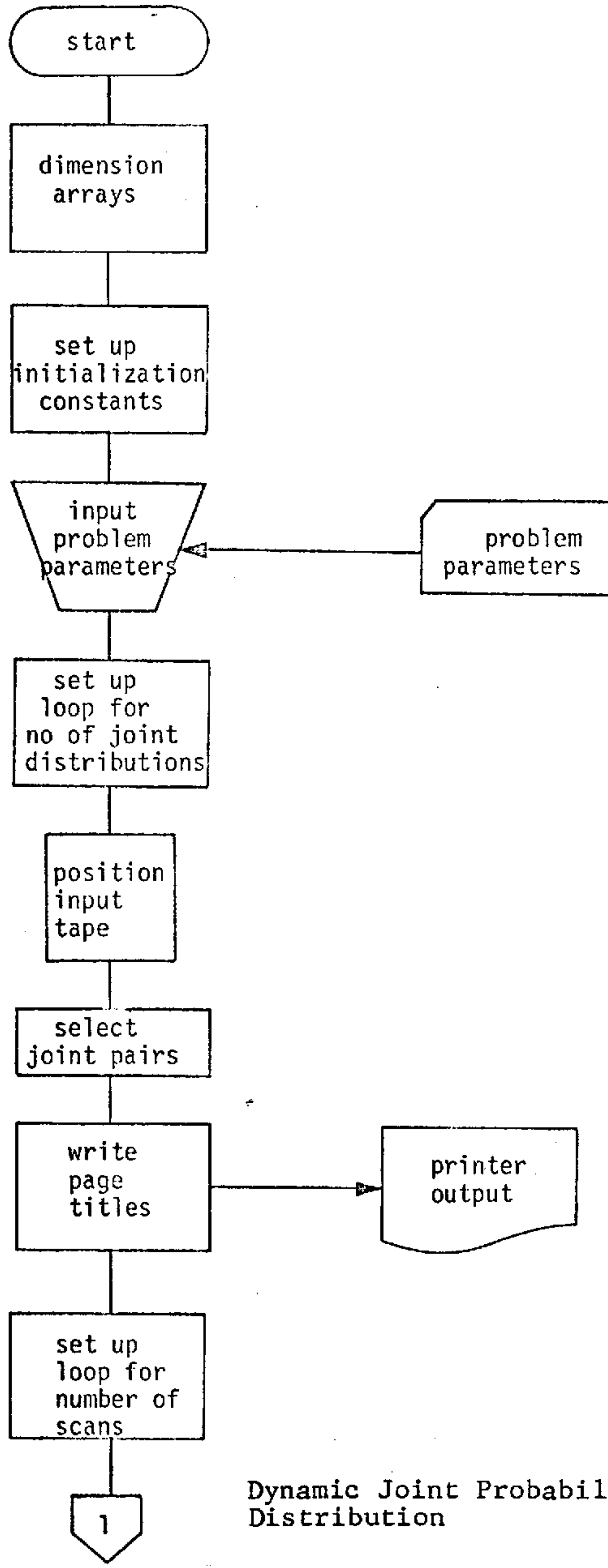
Unit - A6 (optional)

Type - Any odd parity binary, 3-bit modulus, fixed point with word lengths $\leq$ 36 bits, or floating point binary.

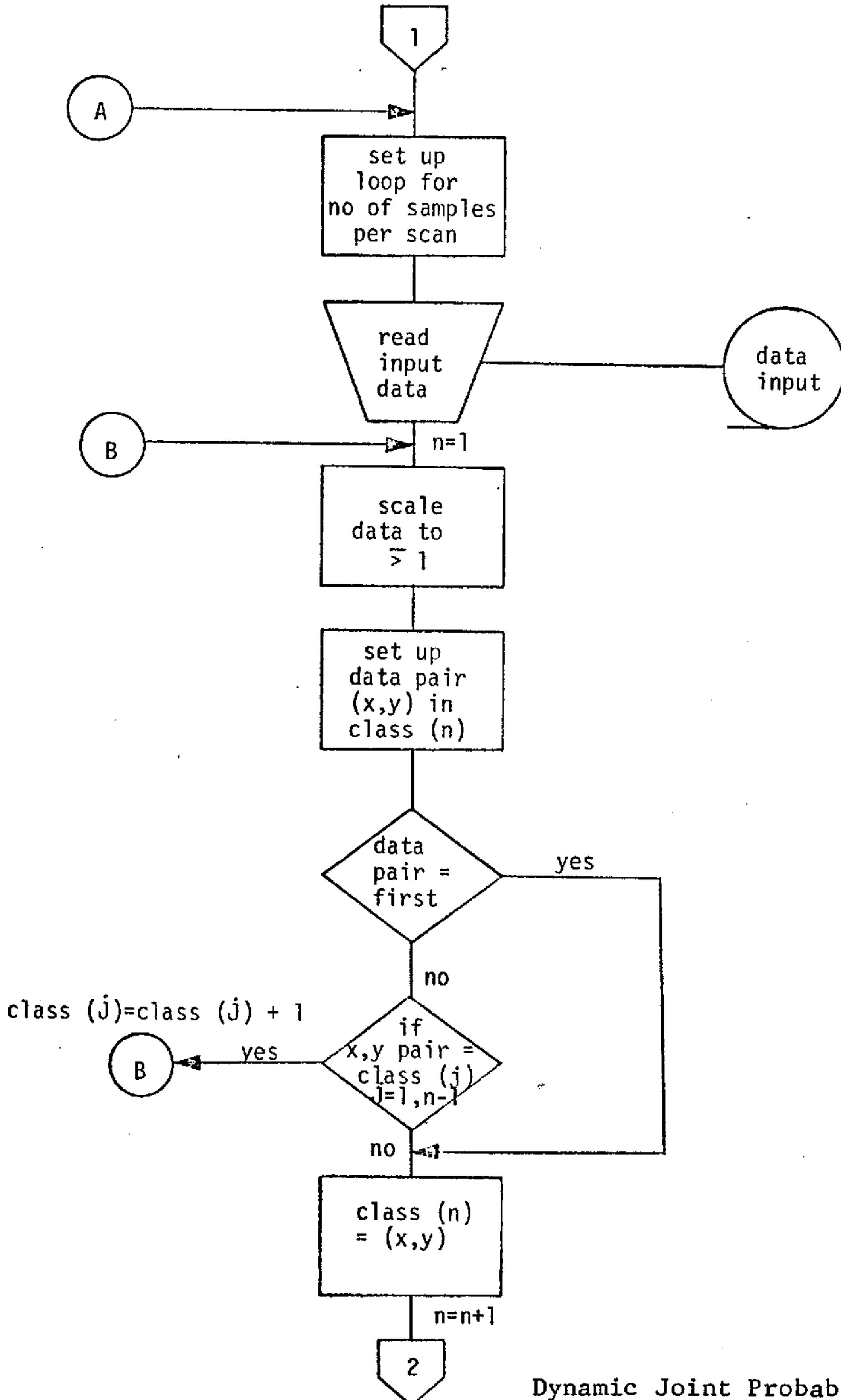
C. Output Tapes

(none print output only)

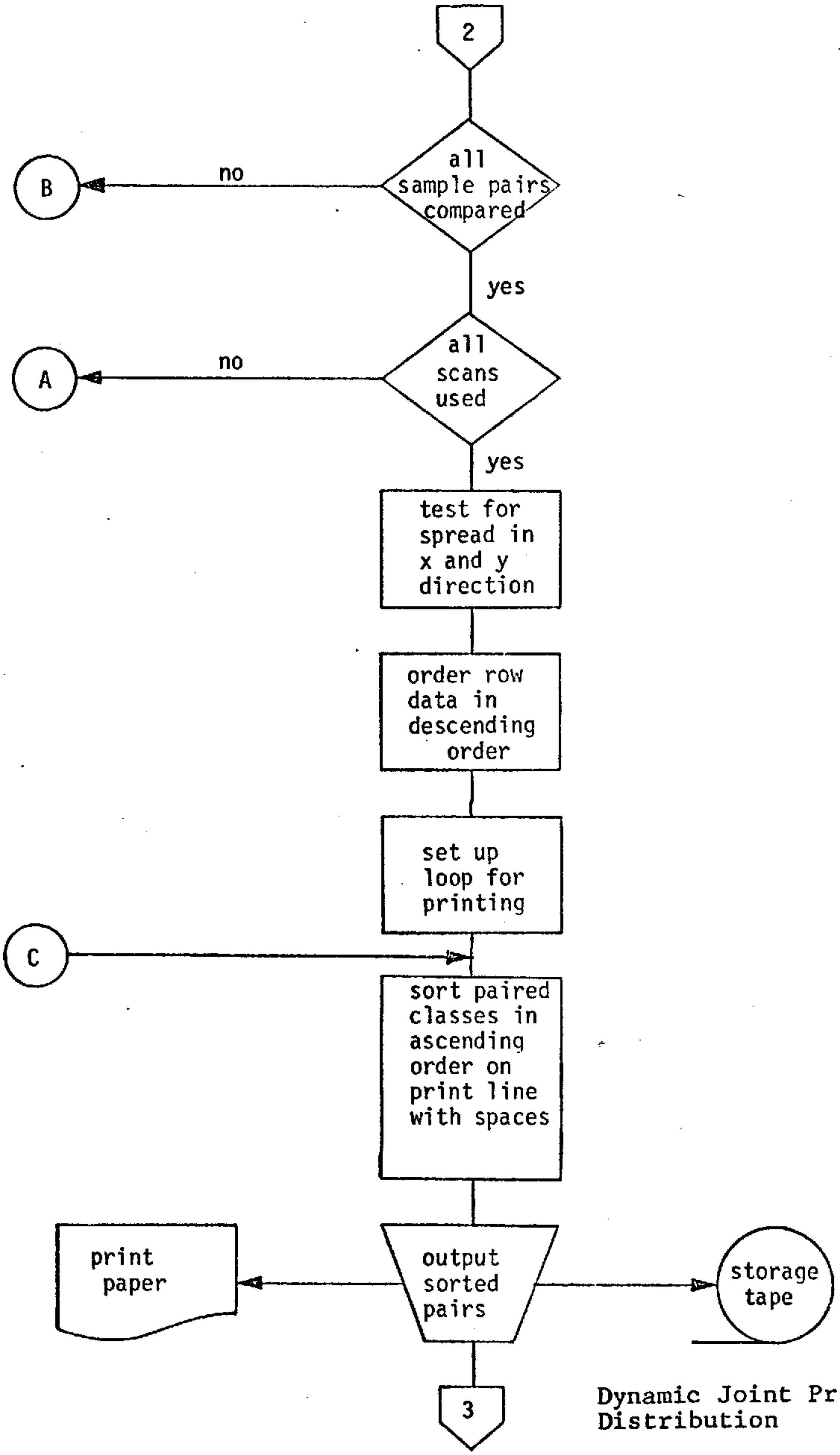
D. Program Flow Chart



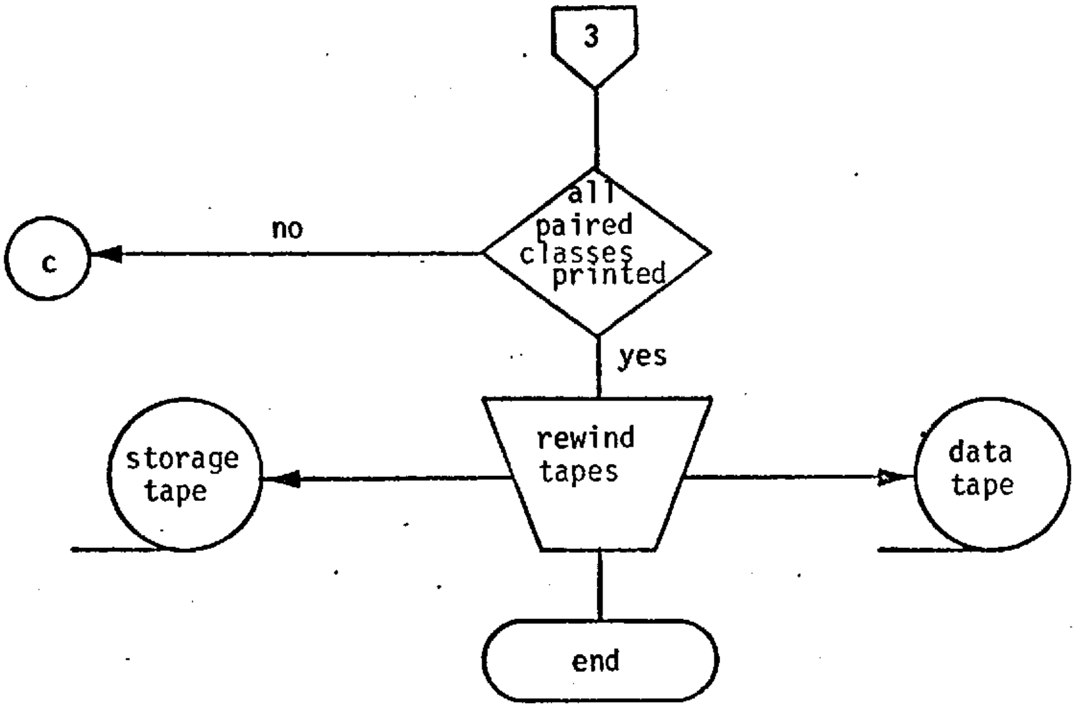
Dynamic Joint Probability Distribution



Dynamic Joint Probability Distribution



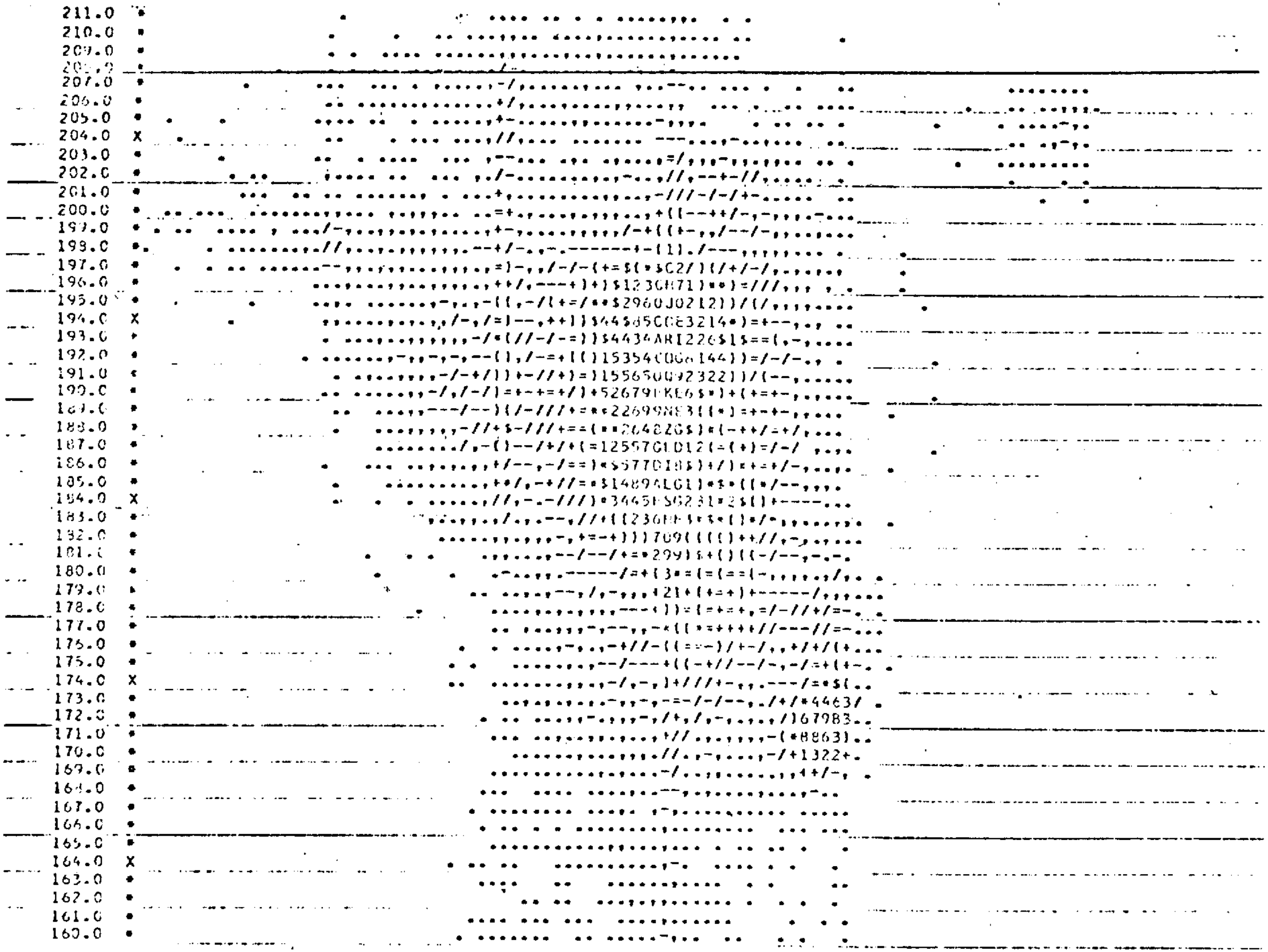
Dynamic Joint Probability Distribution



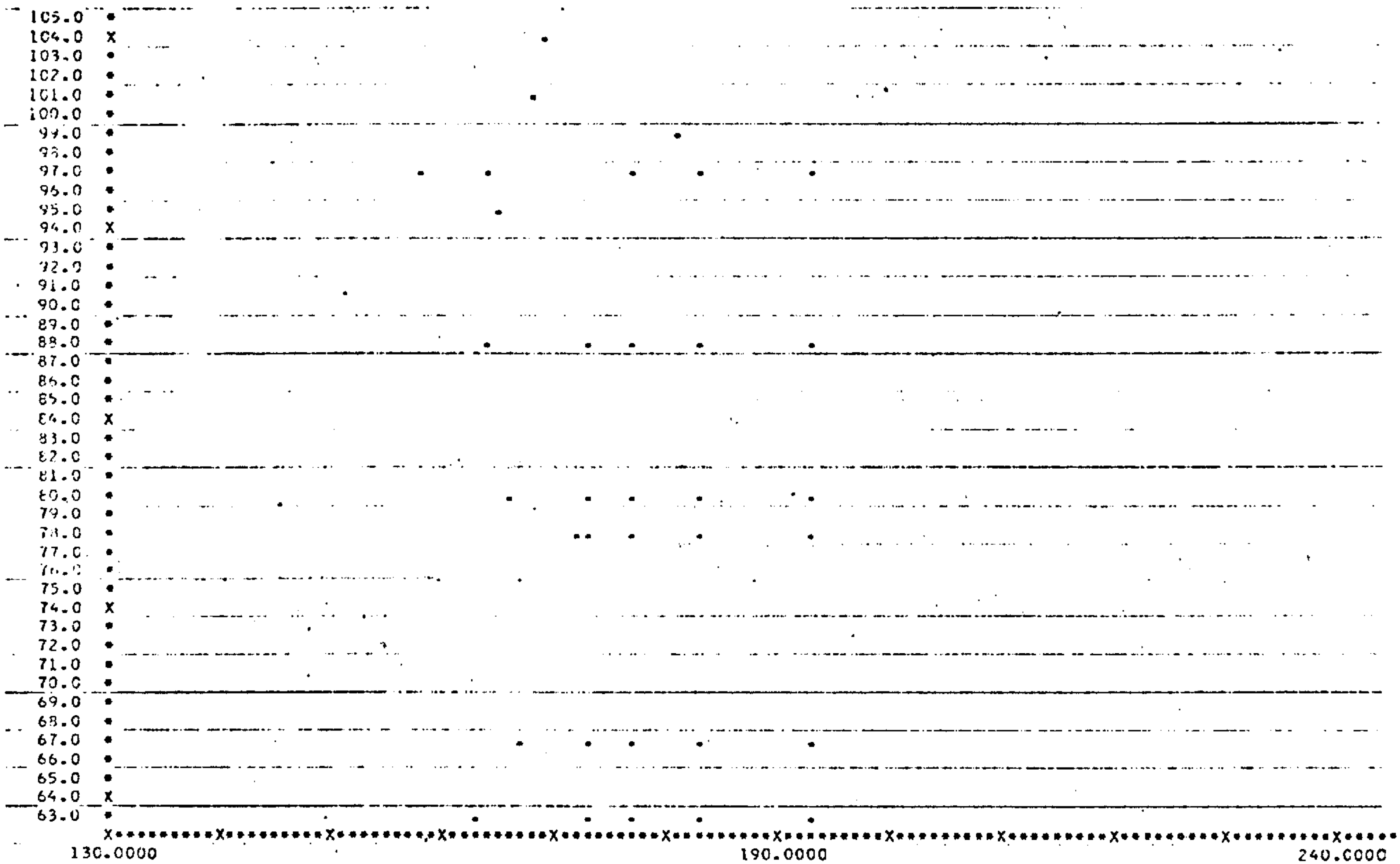
Dynamic Joint Probability Distribution (concluded)

OUTPUT EXAMPLE

```
                    JOINT PROBABILITY DISTRIBUTION
          X-AXIS IS       9        Y-AXIS IS      12
                    DATA SWITCH HAS OCCURRED
          X-AXIS IS      12        Y-AXIS IS       9
SYMBOL       N/SYMBOL
                 0
   .             3
   :             6
   -             9
   /            12
   +            15
   =            18
   (            21
   )            24
   *            27
   $            30
   1            33
   2            36
   3            39
   4            42
   5            45
   6            48
   7            51
   8            54
   9            57
   0            60
   A            63
   B            66
   C            69
   D            72
   E            75
   F            78
   G            81
   H            84
   I            87
   J            90
   K            93
   L            96
   M            99
   N           102
   O           105
   P           108
   Q           111
   R           114
   S           117
   T           120
   U           123
   V           126
   W           129
   X           132
   Y           135
   Z           138
        *******************************************************************
        *                          PART 1 OF 1
        *******************************************************************
   214.0 X
   213.0 *
   212.0 *
```


211.0
210.0
209.0
207.0
206.0
205.0
204.0
203.0
202.0
201.0
200.0
198.0
197.0
196.0
195.0
194.0
193.0
192.0
191.0
188.0
187.0
186.0
185.0
183.0
182.0
180.0
179.0
178.0
177.0
176.0
175.0
174.0
173.0
172.0
171.0
170.0
169.0
167.0
166.0
165.0
164.0
163.0
162.0
161.0
160.0


159.0
158.0
157.0
156.0
155.0
154.0 X
153.0
152.0
151.0
150.0
149.0
148.0
147.0
146.0
145.0
144.0 X
143.0
142.0
141.0
140.0
139.0
138.0
137.0
136.0
135.0
134.0 X
133.0
132.0
131.0
130.0
129.0
128.0
127.0
126.0
125.0
124.0 X
123.0
122.0
121.0
120.0
119.0
118.0
117.0
116.0
115.0
114.0 X
113.0
112.0
111.0
110.0
109.0
108.0
107.0
106.0


105.0
104.0
103.0
102.0
101.0
100.0
99.0
98.0
97.0
96.0
95.0
94.0
93.0
92.0
91.0
90.0
89.0
88.0
87.0
86.0
85.0
84.0
83.0
82.0
81.0
80.0
79.0
78.0
77.0
76.0
75.0
74.0
73.0
72.0
71.0
70.0
69.0
68.0
67.0
66.0
65.0
64.0
63.0
130.0000
190.0000
240.0000

7. MODULE SIX

Module six calculates boundaries outlining edges of homogeneous areas of ground scene data. These boundaries are displayed on print paper where a selected alphanumeric character signifies a boundary and blanks signify areas of homogeneity. This output is also displayed on the SC4020 plotter where boundaries are flagged by an alphanumeric "period." The boundary output is used in further processing to extract information for feature discrimination.

A. Data Problem Parameters Example Setup

$INPUT6

NSCANS = 120	Process 120 scans of the data set
NSTART = 1	Starting sample number
NSPS = 222	Number of samples to process; ($\leq$ 255)
NCH = 12	Number of channels on input tape
NSYM = 49	Dimension of the Alphanumeric array
ISUM = 5	Number of channels used in calculating boundaries
NBTLG = 12	Bit length of input data word
MODE = 1	FORTRAN formatted (1), non-FORTRAN formatted = 2
ITYPE = 0	Input data fixed point. Floating point binary = 1
MSFC = 0	Not MSFC scanner format MSFC scanner format = 1
NSKIP = 0	Skip no records before processing
INCX = 0	Incrementation in the X direction on the plot frame between resolution elements for each scan
INCY = 4	Incrementation in the Y direction on the plot frame between resolution elements for each scan
NSTX = 0	Starting X coordinates on the plot frame
NSTY = 0	Starting Y coordinates on the plot frame
NCRE = 2	Data incrementation; Use every other data sample for boundary calculations

$END

$NCHUSE

NWHICH = 1,3,4,8,12 Channel selection for calculating boundaries

$END

1. 3. 1 *(- =, $+ABC ....Z12 ...90-
7/8

B. Input Tapes

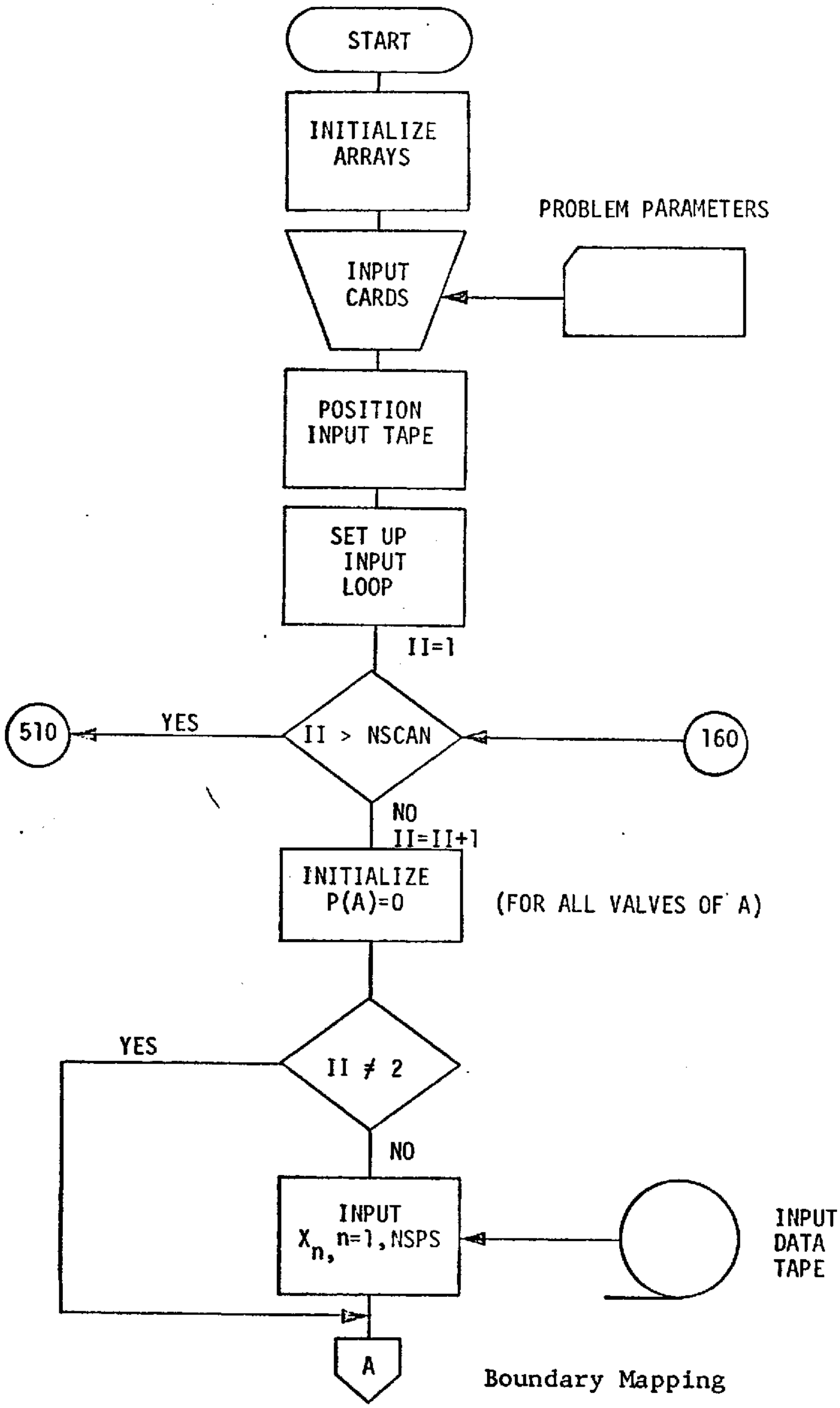
Unit - Any odd parity binary, 3-bit modulus, fixed point with word lengths $\leq$ 36 bits, or floating point binary.

C. Output Tapes

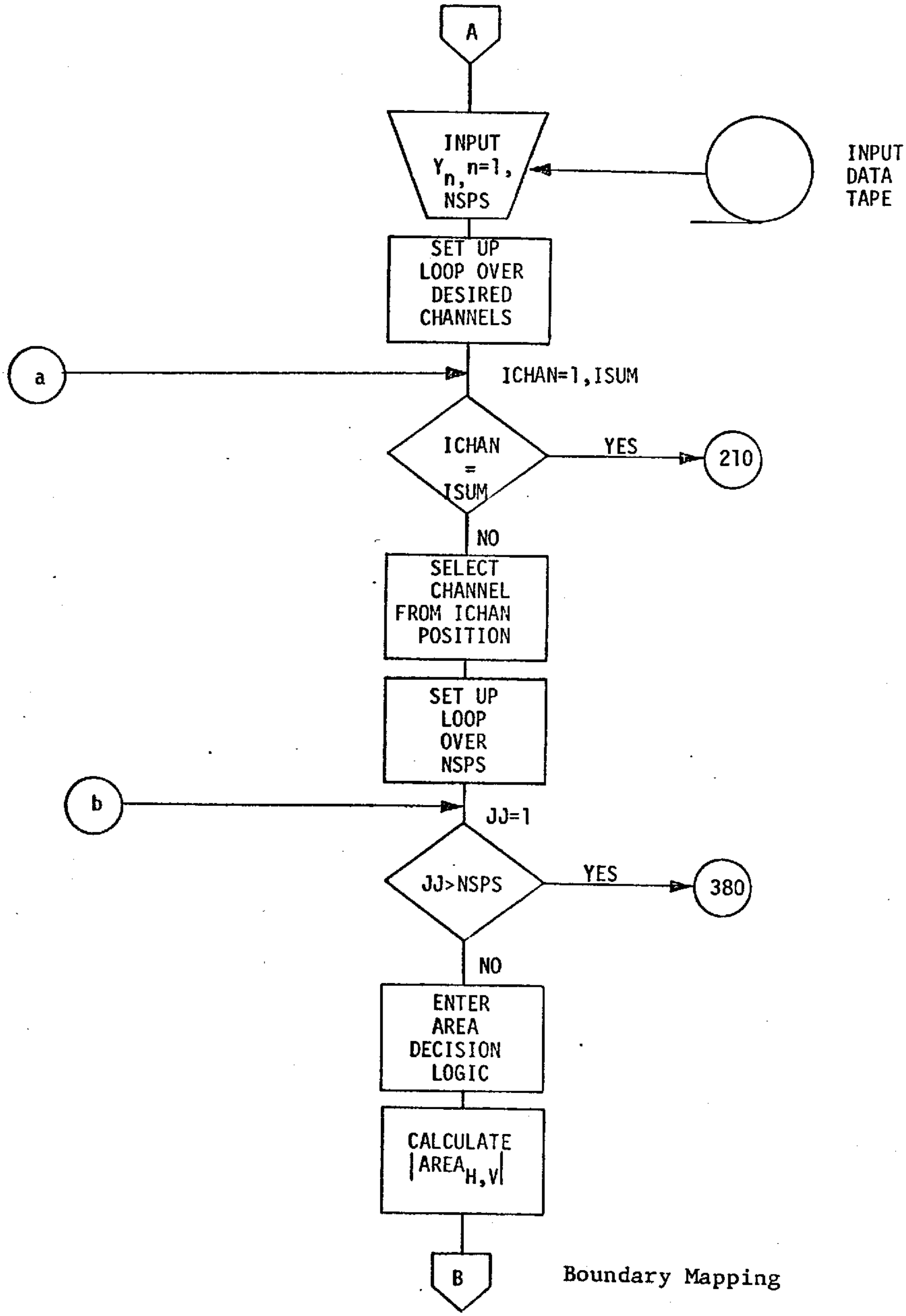
Unit - B6

Type - Fixed point binary 1's and 0's FORTRAN formatted 255 words or less plus 1 FORTRAN index per logical record.

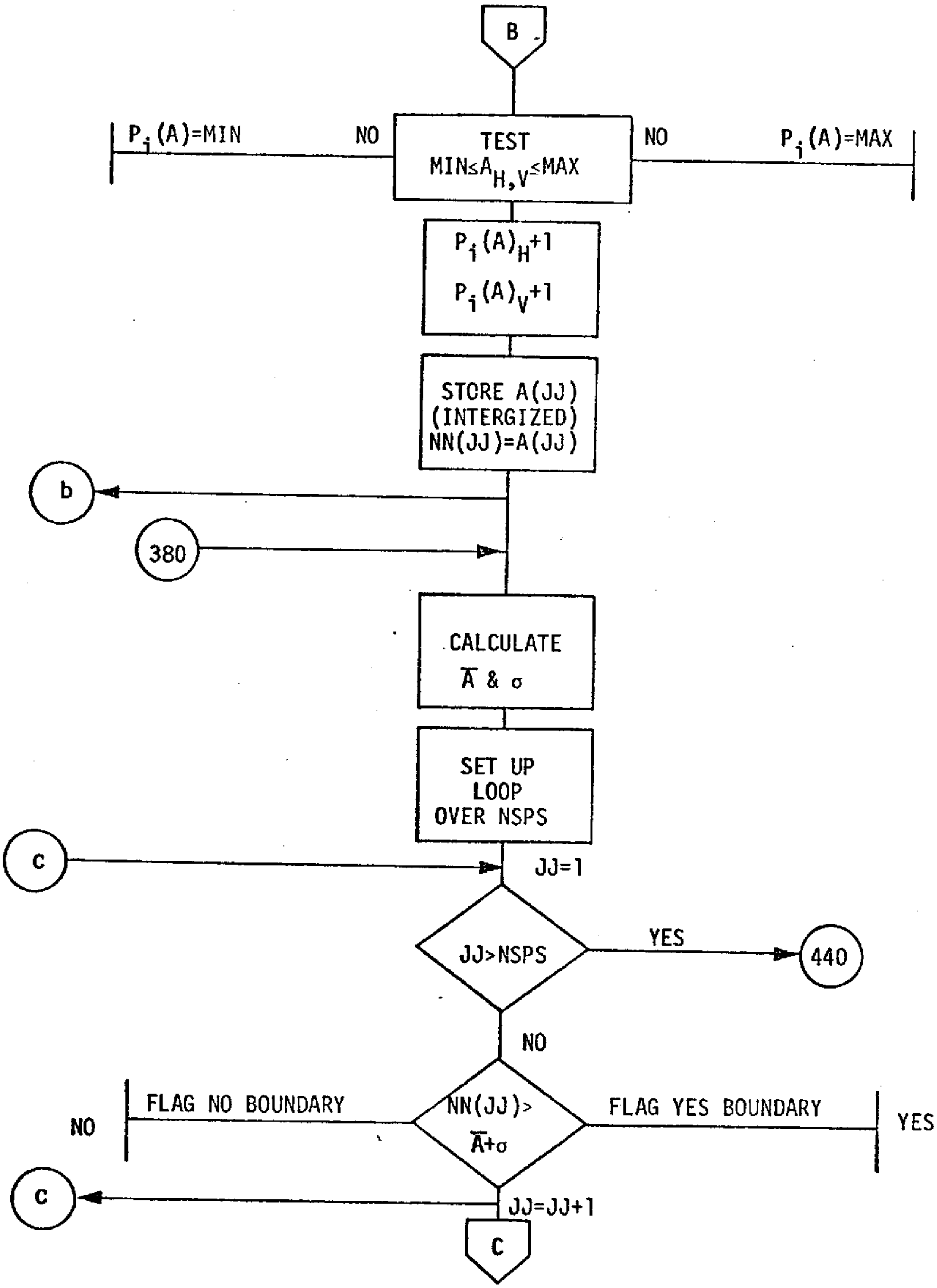
D. Program Flow Chart



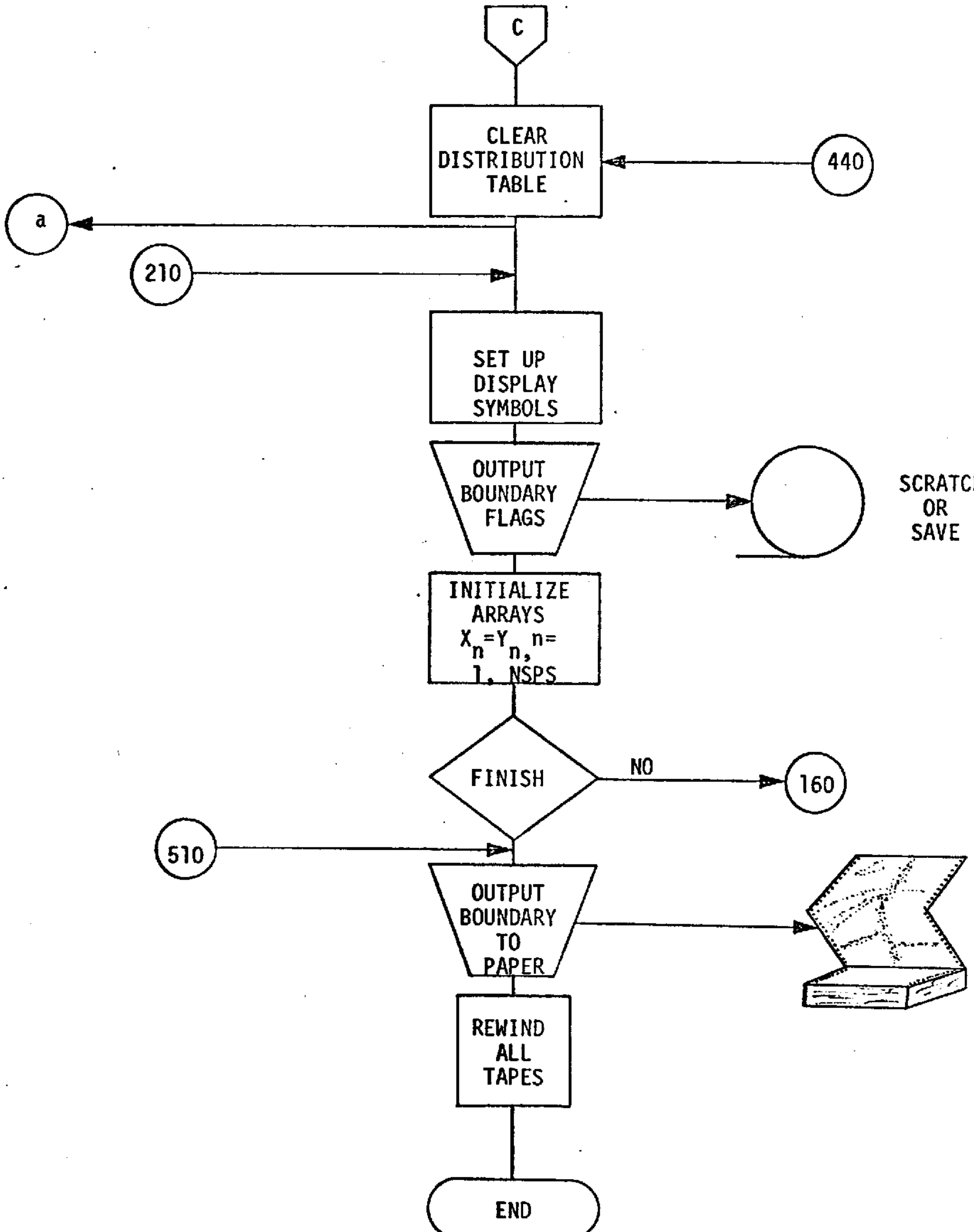
Boundary Mapping



Boundary Mapping



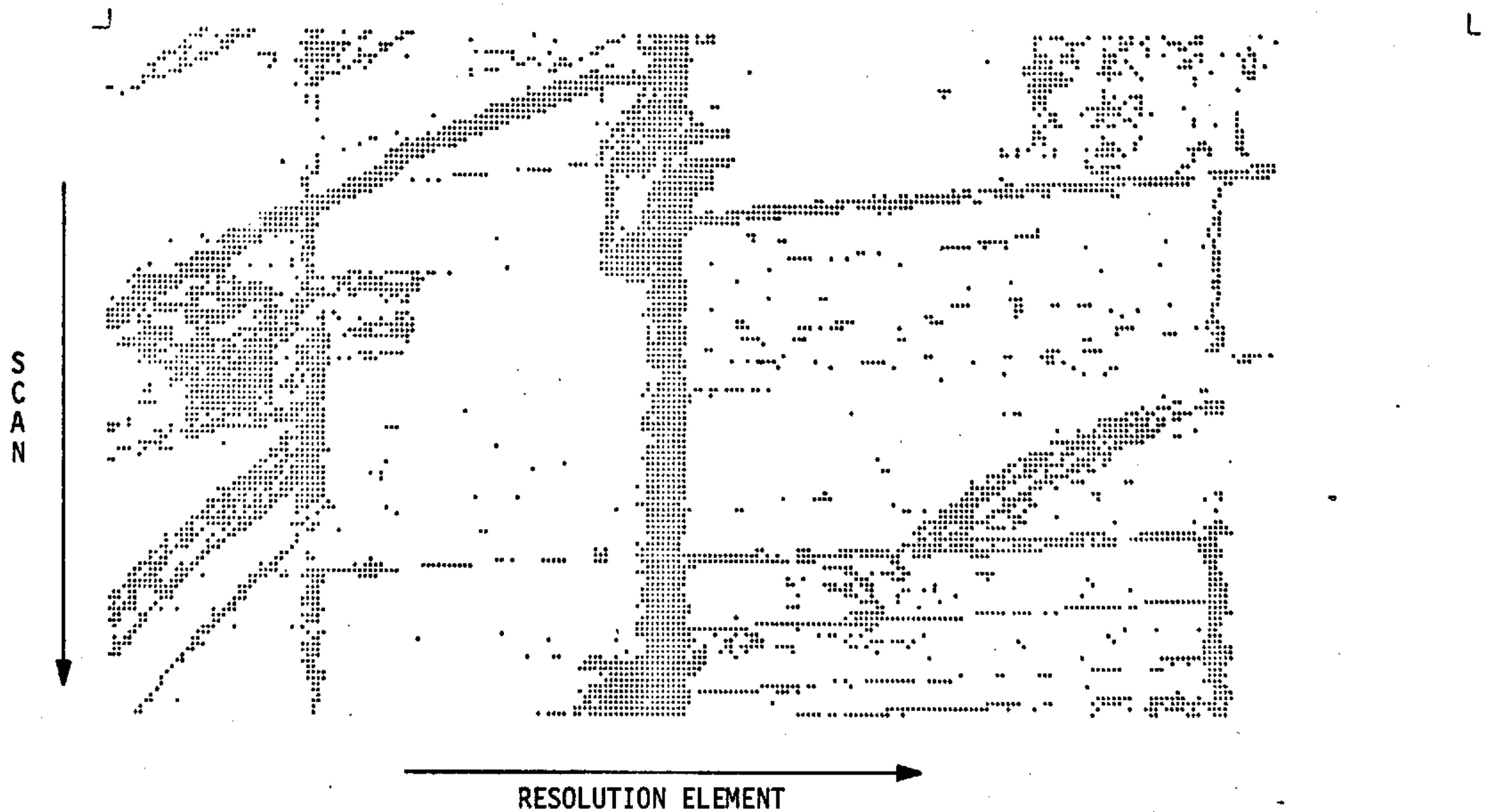
Boundary Mapping



Boundary Mapping (concluded)

OUTPUT EXAMPLE

SCAN 1-120 RESOLUTION ELEMENT 1-222



Boundary Determination Channels 4 of 12 (Purdue C1)

8. MODULE SEVEN

Module seven consists of three submodules called into core memory sequentially to classify land use data. These modules perform spatial clustering, sequential merging, and spectral discrimination of ground scene images. The module provides an option to select multiple passes to further classify small insignificant homogeneous areas that may be overlooked or misclassified during the first pass.

The input to module seven is the boundary mapped information (see Boundary Mapping) and the raw spectral digitized images. The spatial clustering submodule searches areas on the boundary map tape that contain homogeneous areas consisting of at least 100 square resolution elements as a threshold. When this search is satisfied, this homogeneous area is identified with a number which is incremented for each separate homogeneous area. A new tape is created containing these identified clusters. The sequential merging submodule searches the tape containing the identified clusters and selects the raw spectral data associated with each cluster. Criterion parameters are calculated and the merging process takes place merging all similar clusters.

The spectral discrimination submodule, classifies the merged clusters starting with an initial class of one, with unit incrementation for each new class detected. The raw spectral data criterion parameters, for each resolution element are compared with all classification criterion parameters. If a comparison exists, then the resolution element becomes classified. If there is no comparison, the resolution element is assigned a blank or a boundary flag, (if the resolution element reflects a boundary).

If the ground scene is not classified satisfactorily, another pass can be performed reducing the homogeneous square area threshold to 36 resolution elements. This will enable smaller homogeneous areas, that were not detected by the 100-square resolution element threshold to be detected and an attempt will be made to classify these areas also.

In the output example of module seven, Figure 25 shows the initial clustering of homogeneous areas that contain at least 100 square resolution elements. Figure 26 shows the results of the sequential merging and the classifying process. In Figure 26 there exist areas that were not classified therefore an additional pass was necessary to improve the classified areas. Figure 27 shows small homogeneous areas clustered using the 36-square resolution element threshold. These areas are clusters O, P, and Q. Figure 28 shows the areas O, P, and Q, have been merged and classified which improves the classified images.

A. Data Problem Parameters Input Example

$INPUT7	
NPASS = 2	Perform two passes through the clustering, merging and classifying process
NCLUST = 0	Initial cluster number. Additional passes can be performed on reruns, by inputing the total number of classes already detected.
$END	
$INPUTA	
NSPS = 222	Number of resolution elements in entire scan or logical record
NSCANS = 120	Number of scans to process
NCH = 12	Total number of channels on input tape
LT1 = 1	Logical tape unit for scratch tape
LT9 = 9	Logical tape unit for storage of update information
LT10 = 10	Logical tape unit for raw data input
LT11 = 11	Logical tape unit for boundary map
LT12 = 12	Logical tape unit for clustered homogeneous areas
LT13 = 13	Logical tape unit containing the final classified map
NSTART = 1	Starting sample number
NSTOP = 140	Stopping sample number; Process only sample number 1-140 (NSTOP-NSTART + 1 $\leq$ 255)
NBTLG = 12	Bit length of word on input raw data tape
MODE = 1	Input data FORTRAN formatted MODE = 2 non-FORTRAN formatted

SPATIAL CLUSTERING

PASS 1 SCAN 1-120 RESOLUTION ELEMENT 1-140

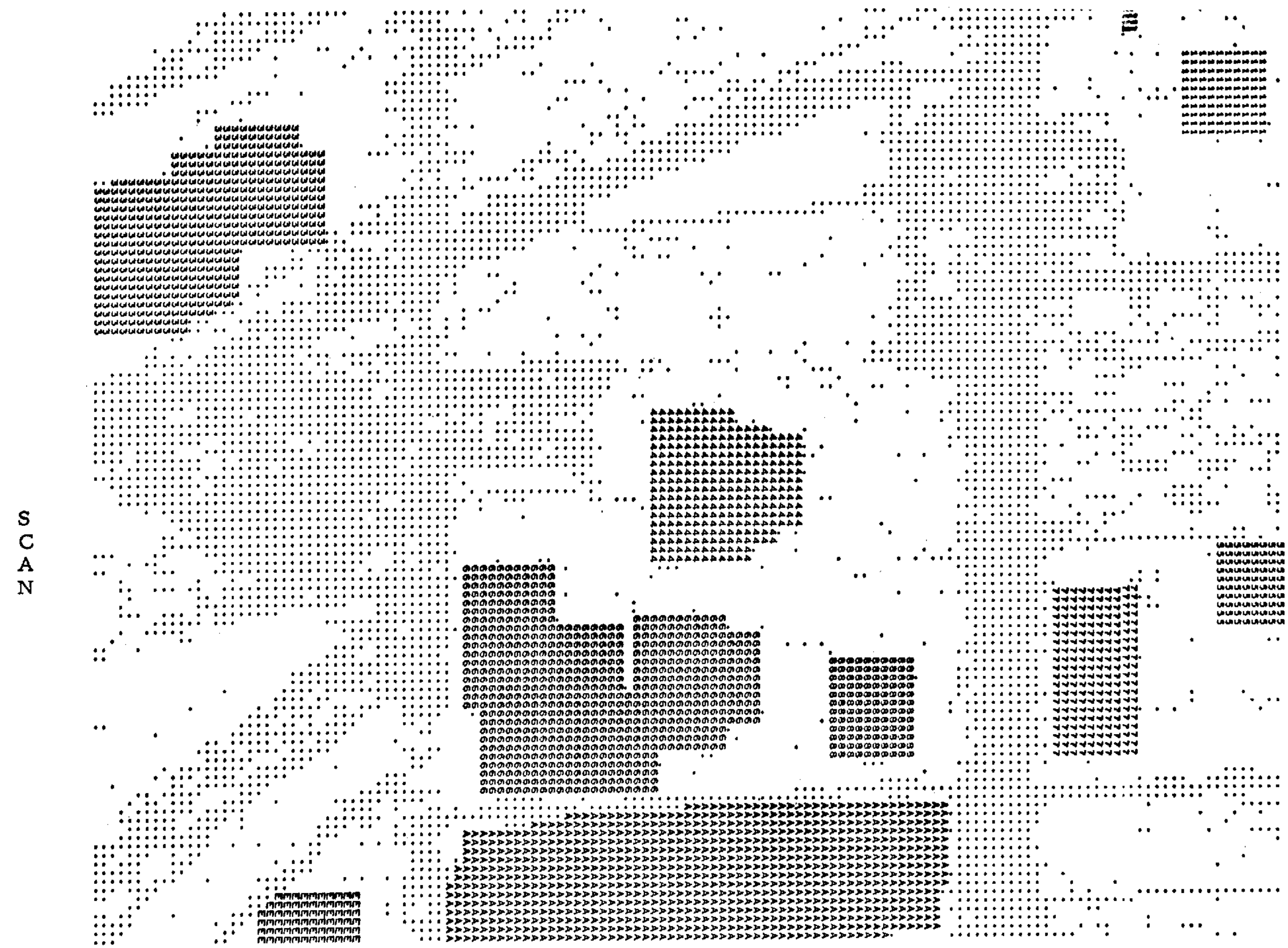
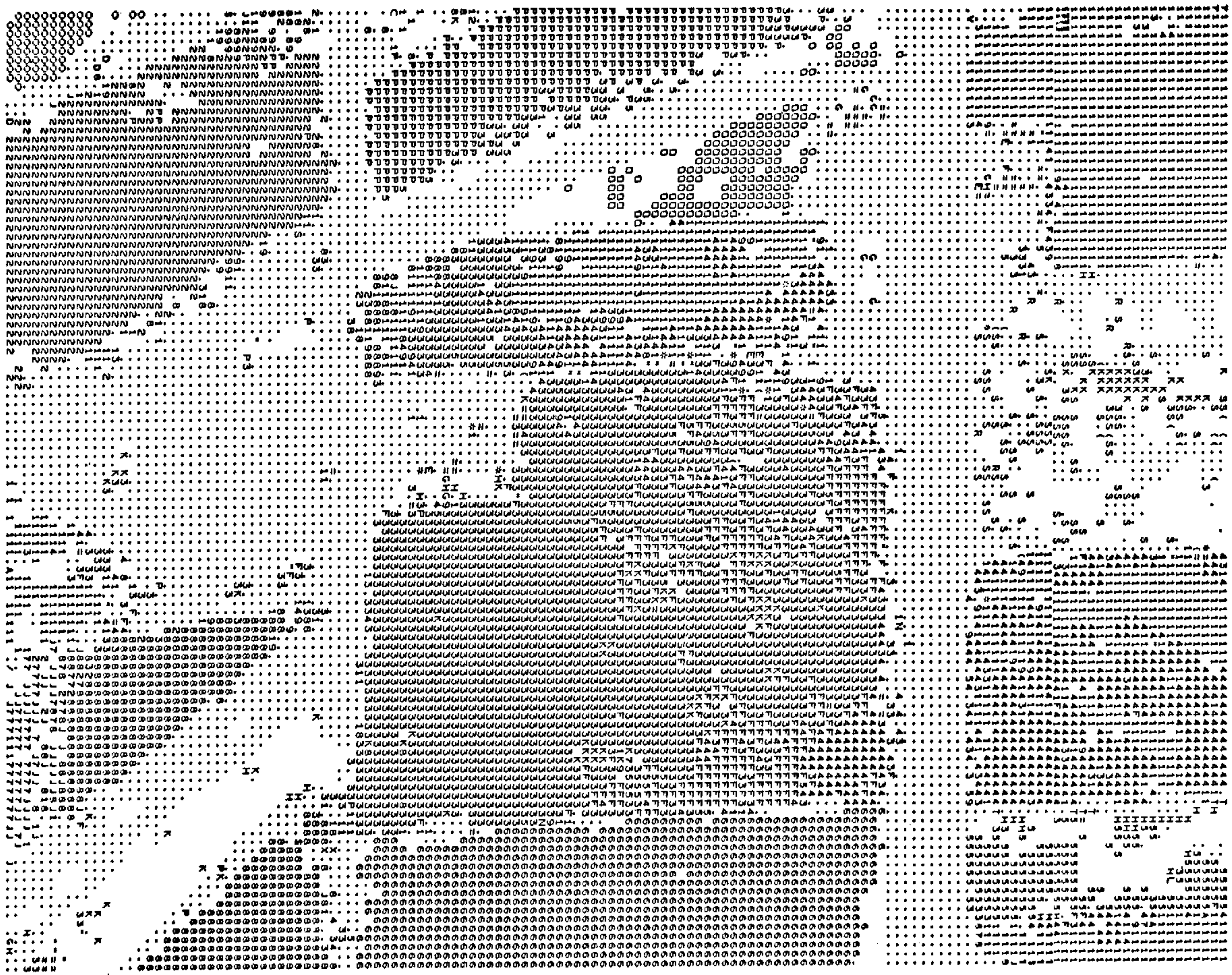


Figure 25 Module Seven Output Example

Figure 26 Results of Sequential Merging
and Classifying Process

Figure 27 Clustered Homogeneous Areas

UNSUPERVISED SPECTRAL DISCRIMINATION
PASS 2 SCAN 1-120 RESOLUTION ELEMENT 1-140



SCAN

Figure 28 Merged and Classified Homogeneous Areas

ITYPE = 0	Input tape binary integer ITYPE = 1, floating point format
MSFC = 0	Non-MSFC scanner format; MSFC scanner has housekeeping data requiring special handling. Set MSFC = 1 if MSFC scanner data
14 = 1	Number of physical records per logical record
NCRE = 1	Data increment; NCRE = 1 use every data sample
NSKIP = 0	Initial records to skip before processing
INCX = 0	Increment in X direction on SC4020 plot frame for each sample
INCY = 8	Increment in Y direction on SC4020 plot frame for each sample
NSTX = 0	Starting X coordinate on the SC4020 plot frame
NSTY = 0	Starting Y coordinate on the SC4020 plot frame
IXXX = 10	Homogeneous area threshold samples in X
IYYY = 10	Homogeneous area threshold samples in Y direction

1 1 1234567890ABC ....(Alphanumeric characters for displaying classified homogeneous areas)

B. Input Tapes

Unit - A6 and B6

A6 is any odd parity binary, 3-bit modulus, fixed point with word lengths $\leq$ 36 bits, or floating point binary.

B6 is integer binary FORTRAN formatted.

C. Output Tapes

Units - B5 FORTRAN floating point binary (update information)

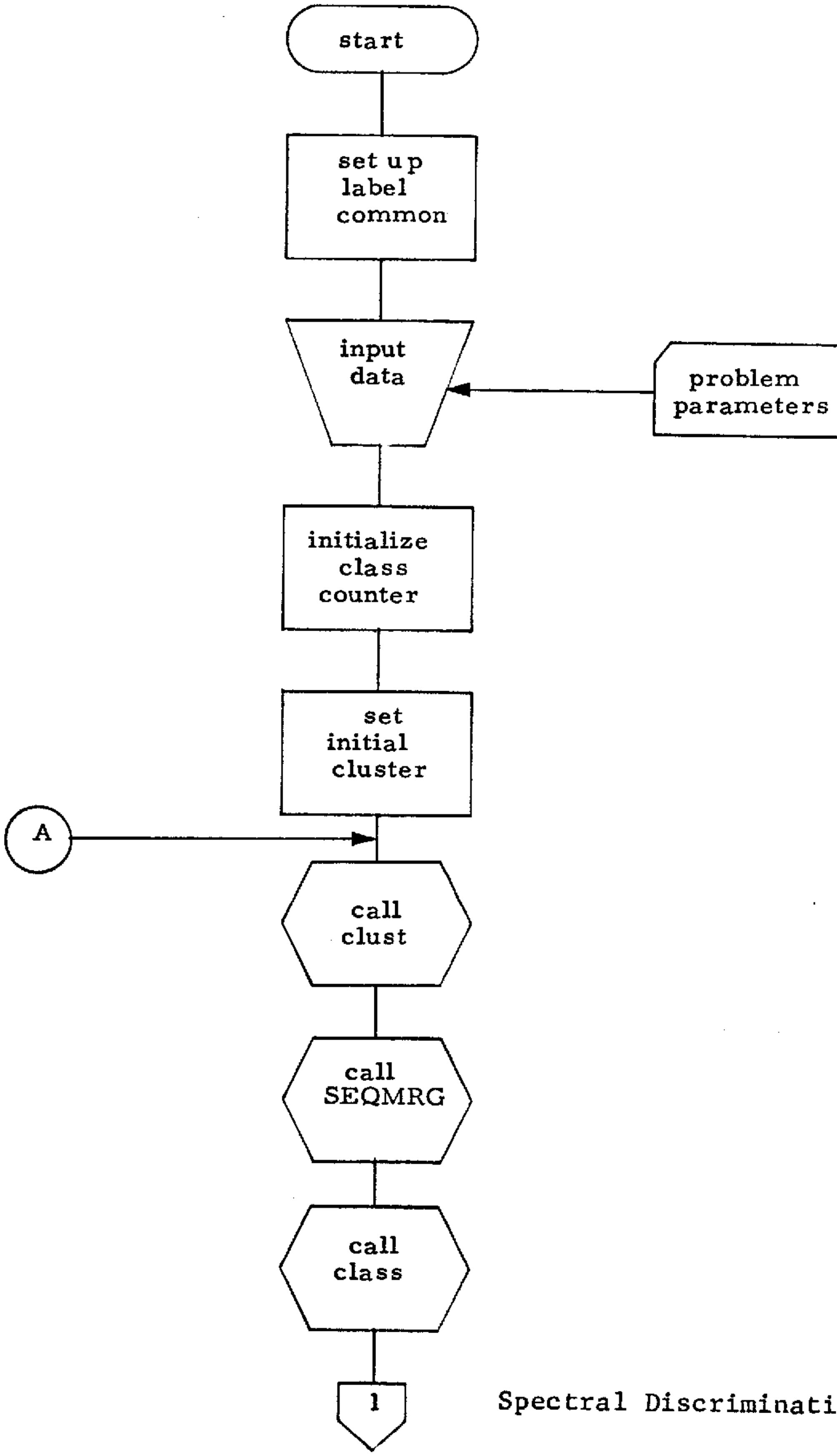
A7 FORTRAN fixed point binary (Cluster data)

B7 FORTRAN fixed point binary

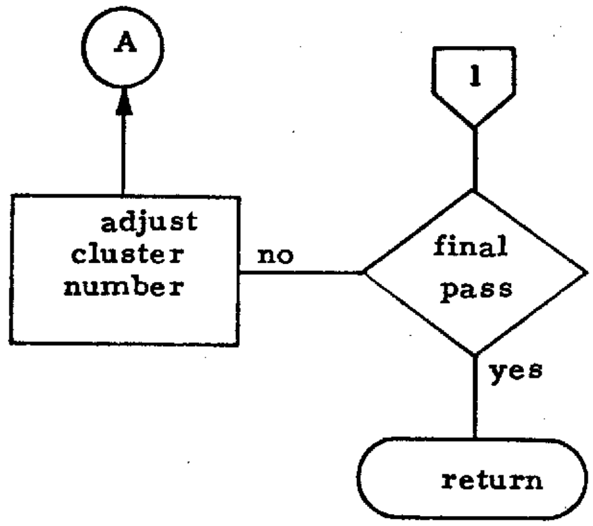
D. Intermediate Tapes

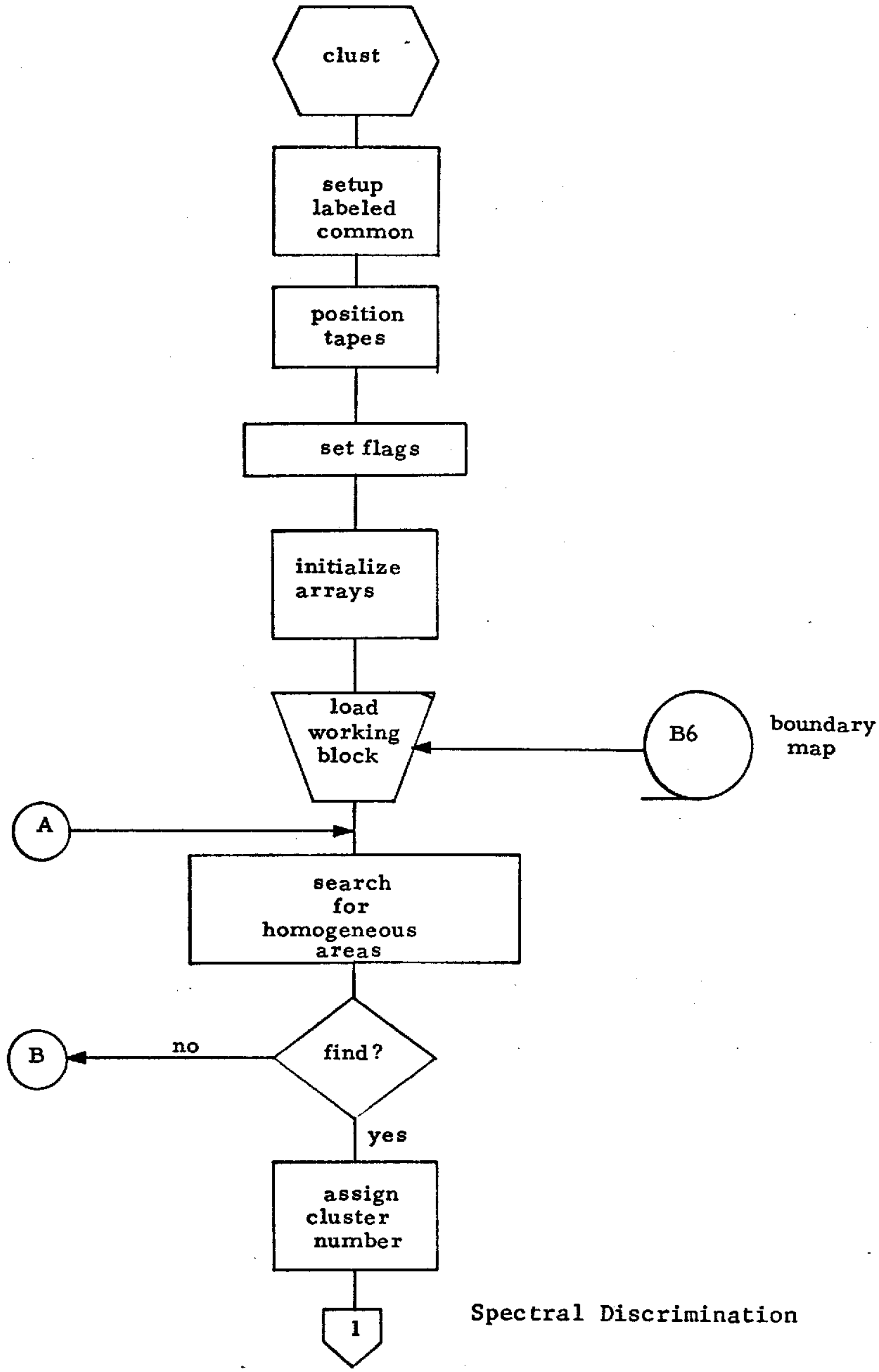
Units - A3, Scratch tape only

E. Program Flow Chart

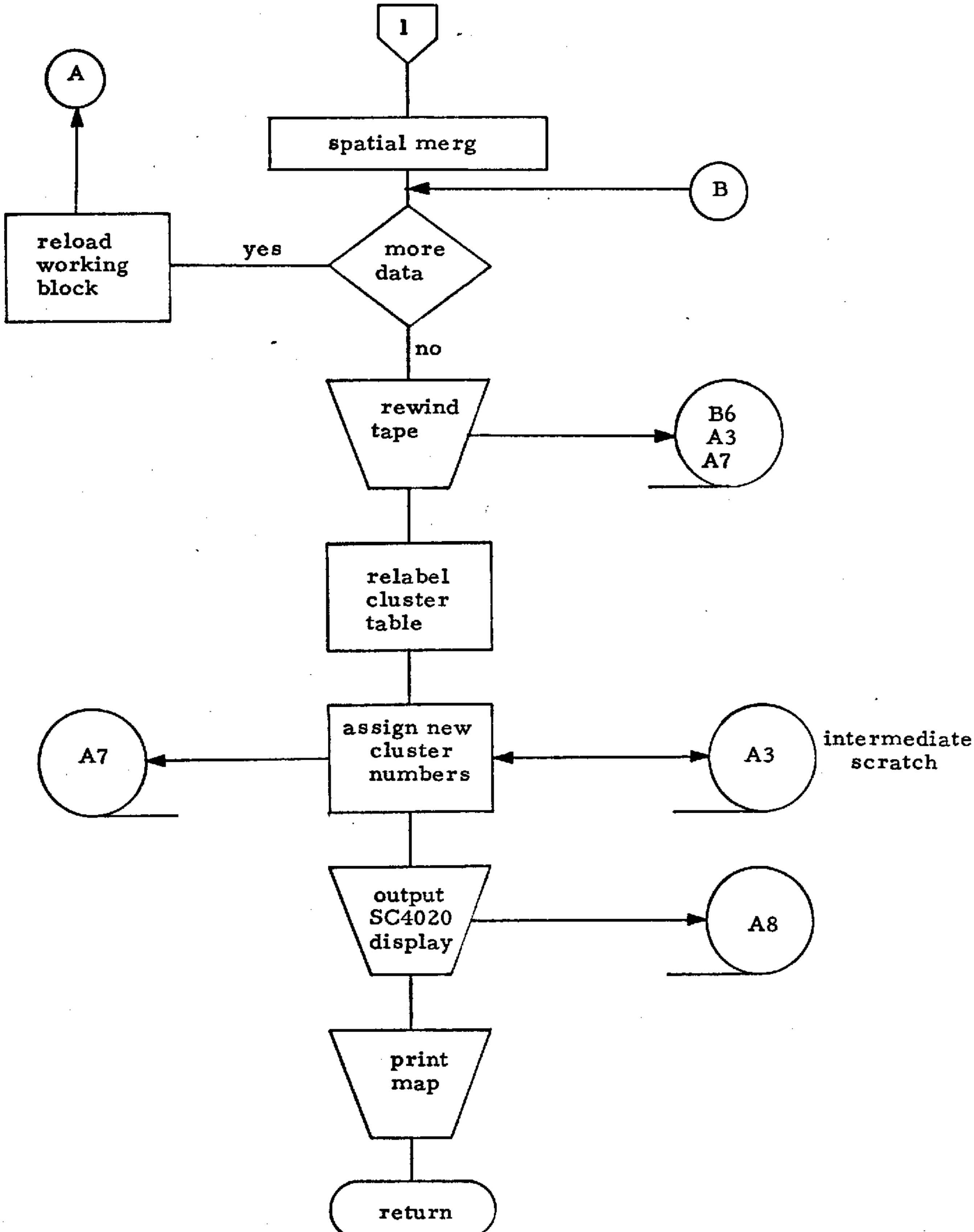


Spectral Discrimination

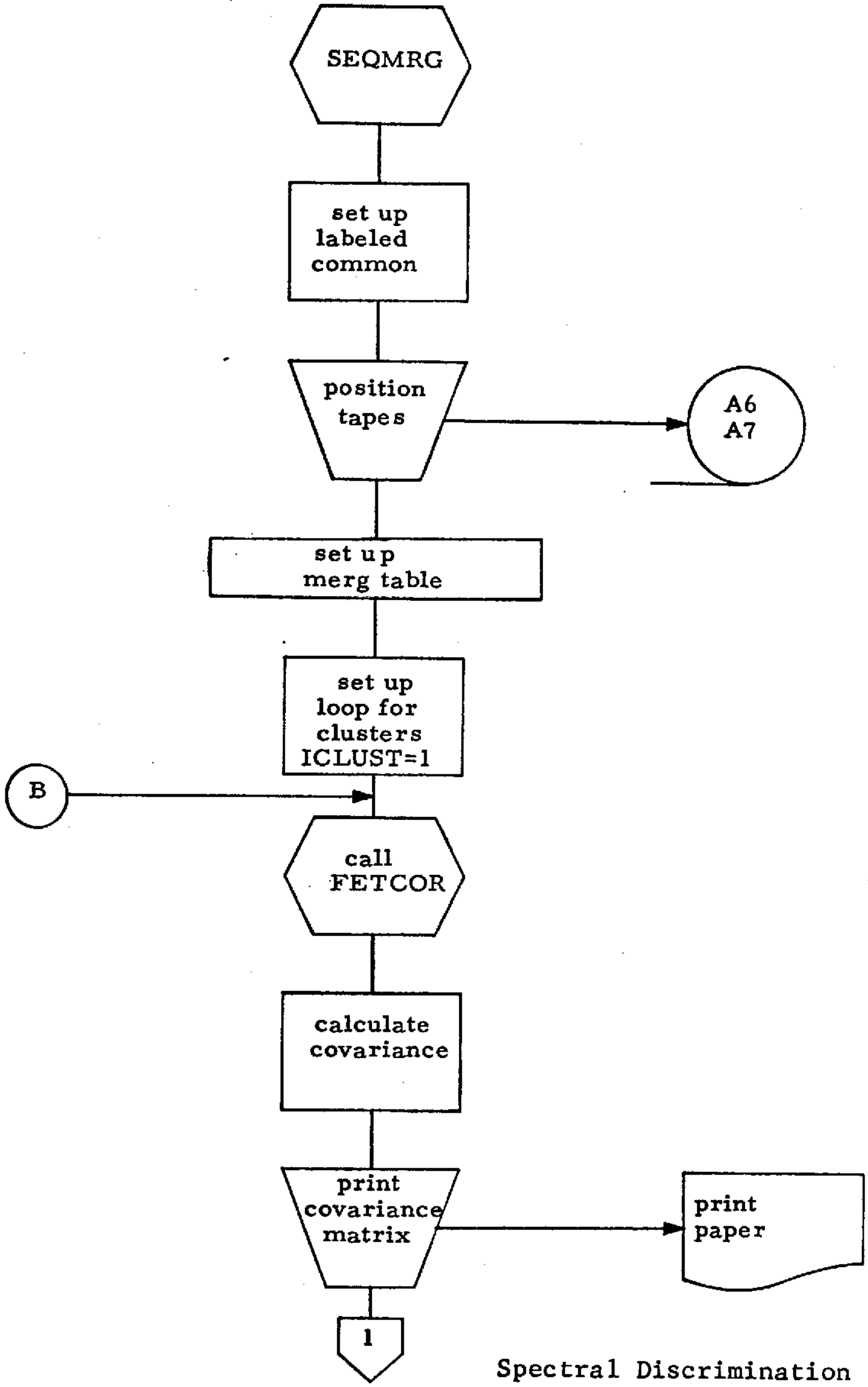

A
1
adjust
cluster
number
no
final
pass
yes
return



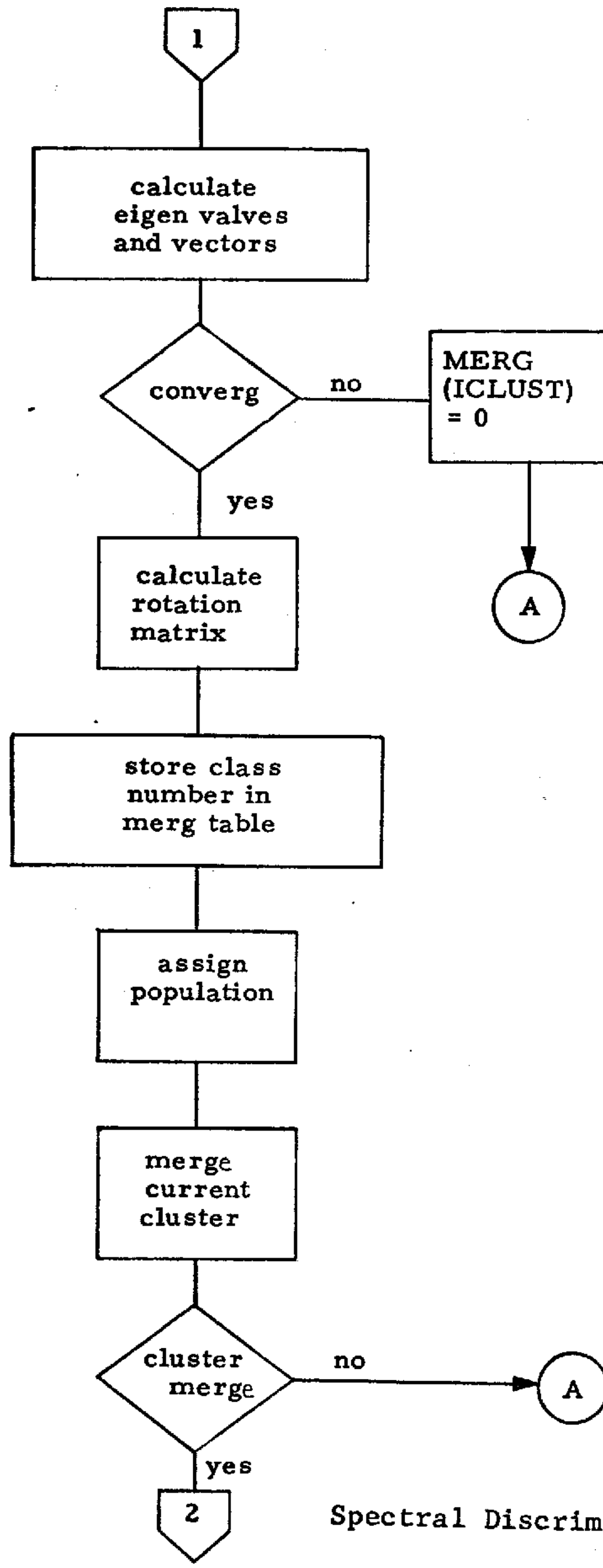
Spectral Discrimination



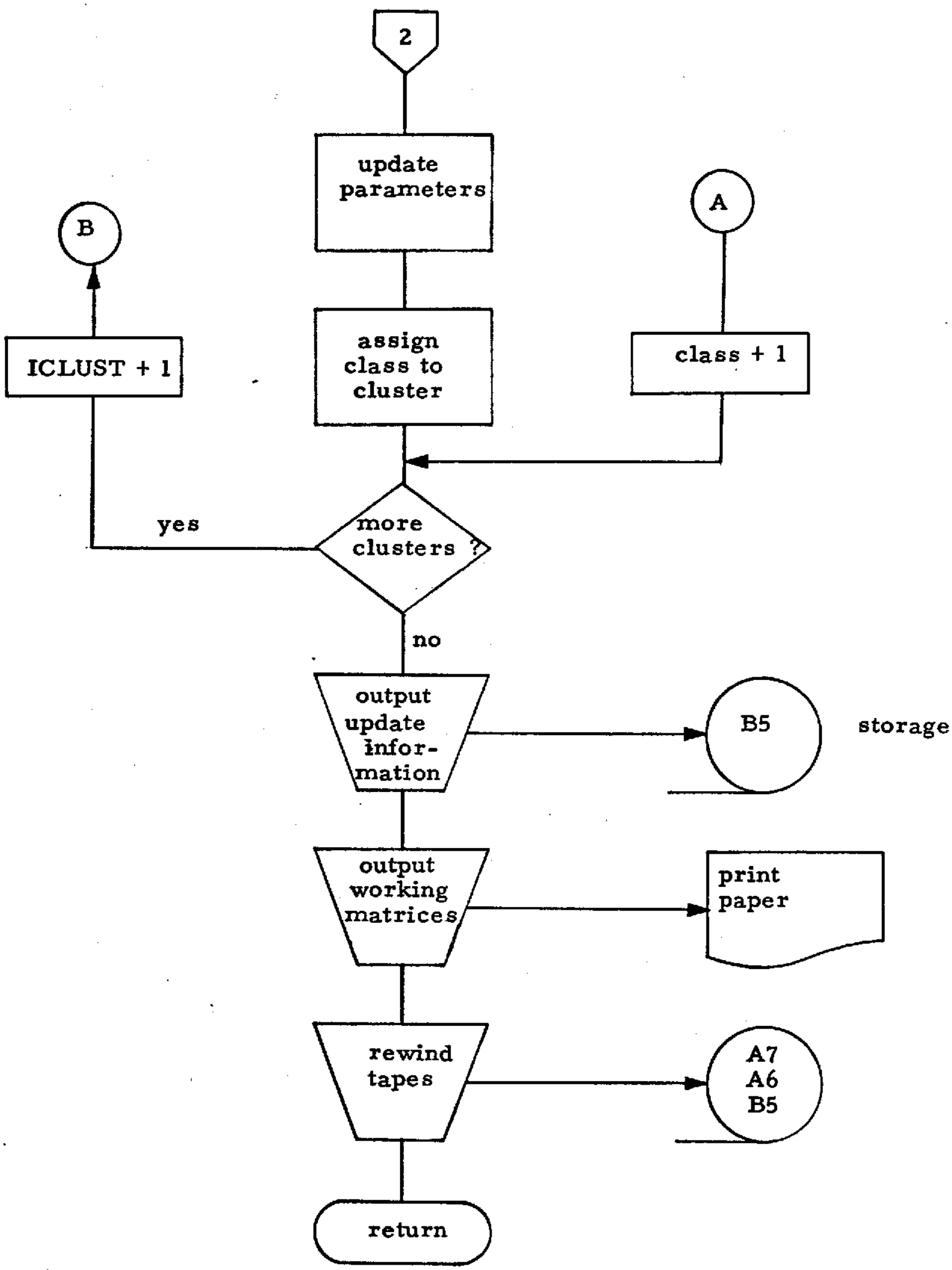
Spectral Discrimination



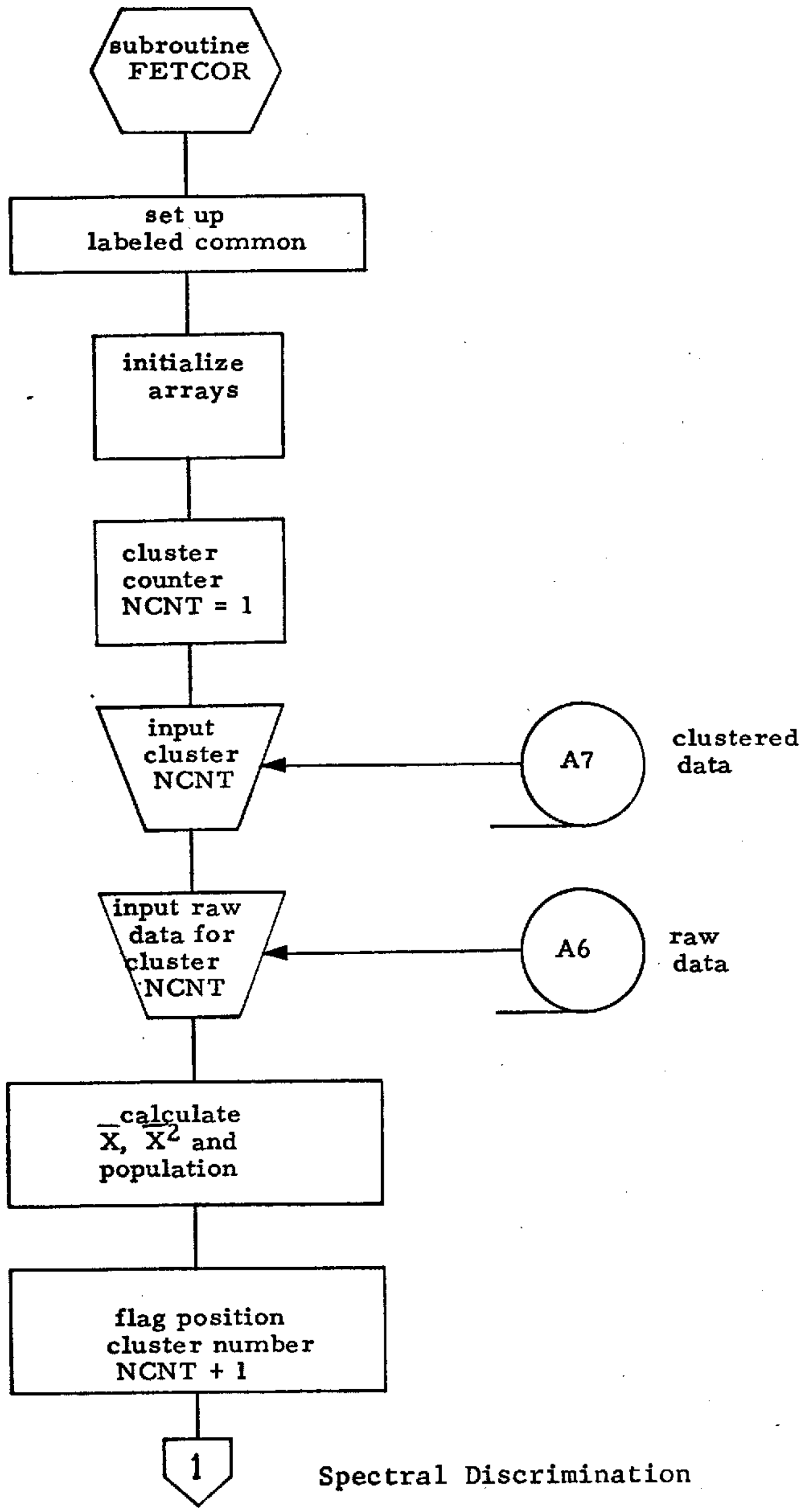
Spectral Discrimination



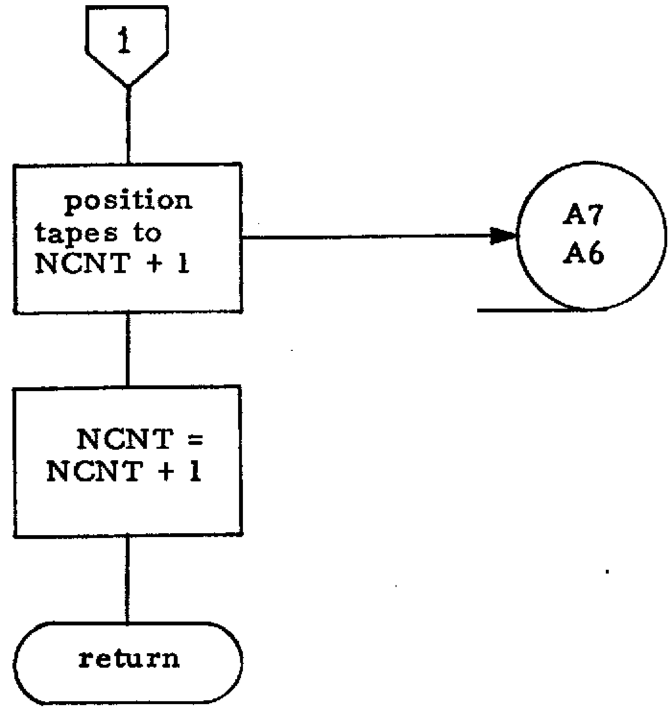
Spectral Discrimination

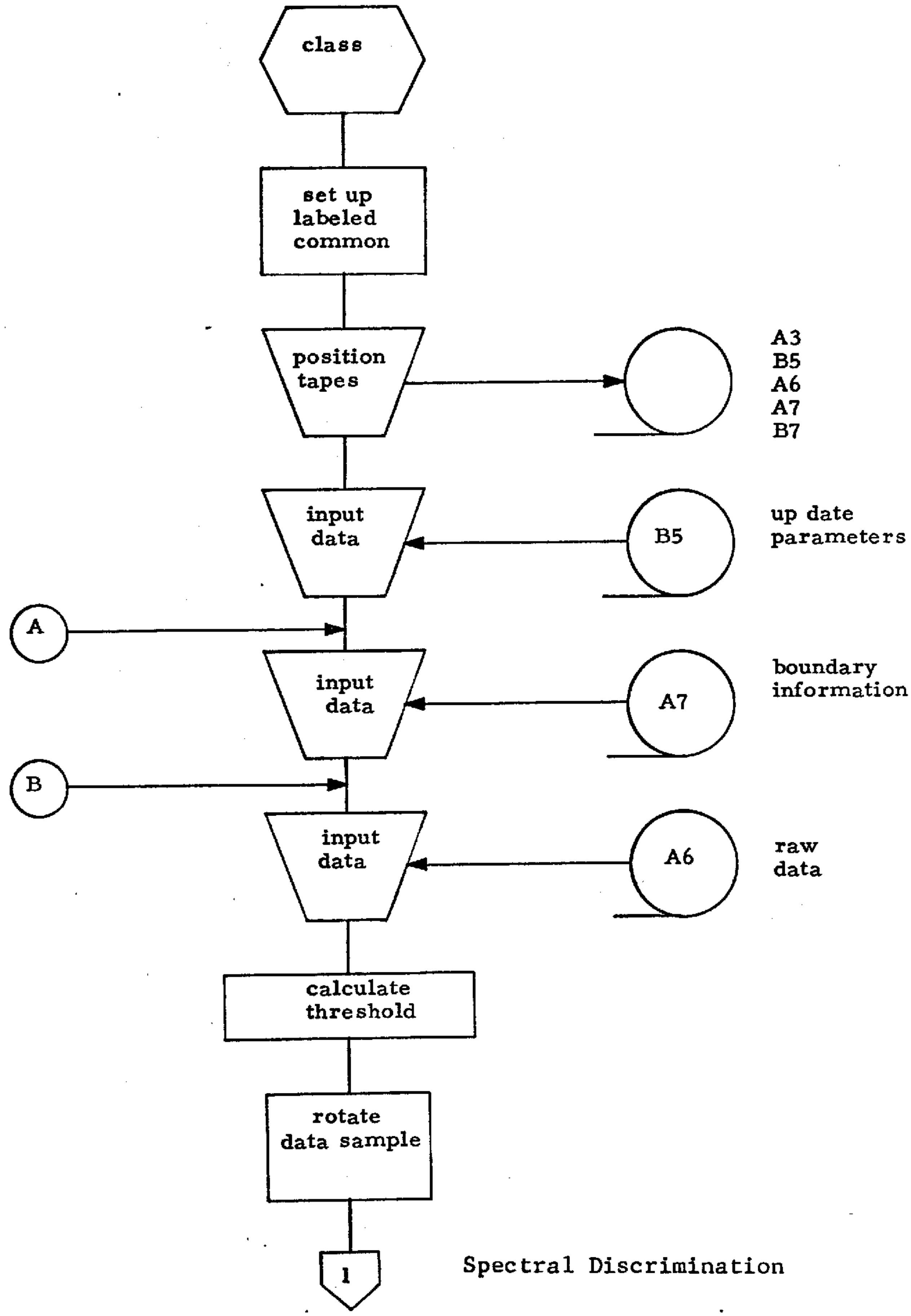


Spectral Discrimination

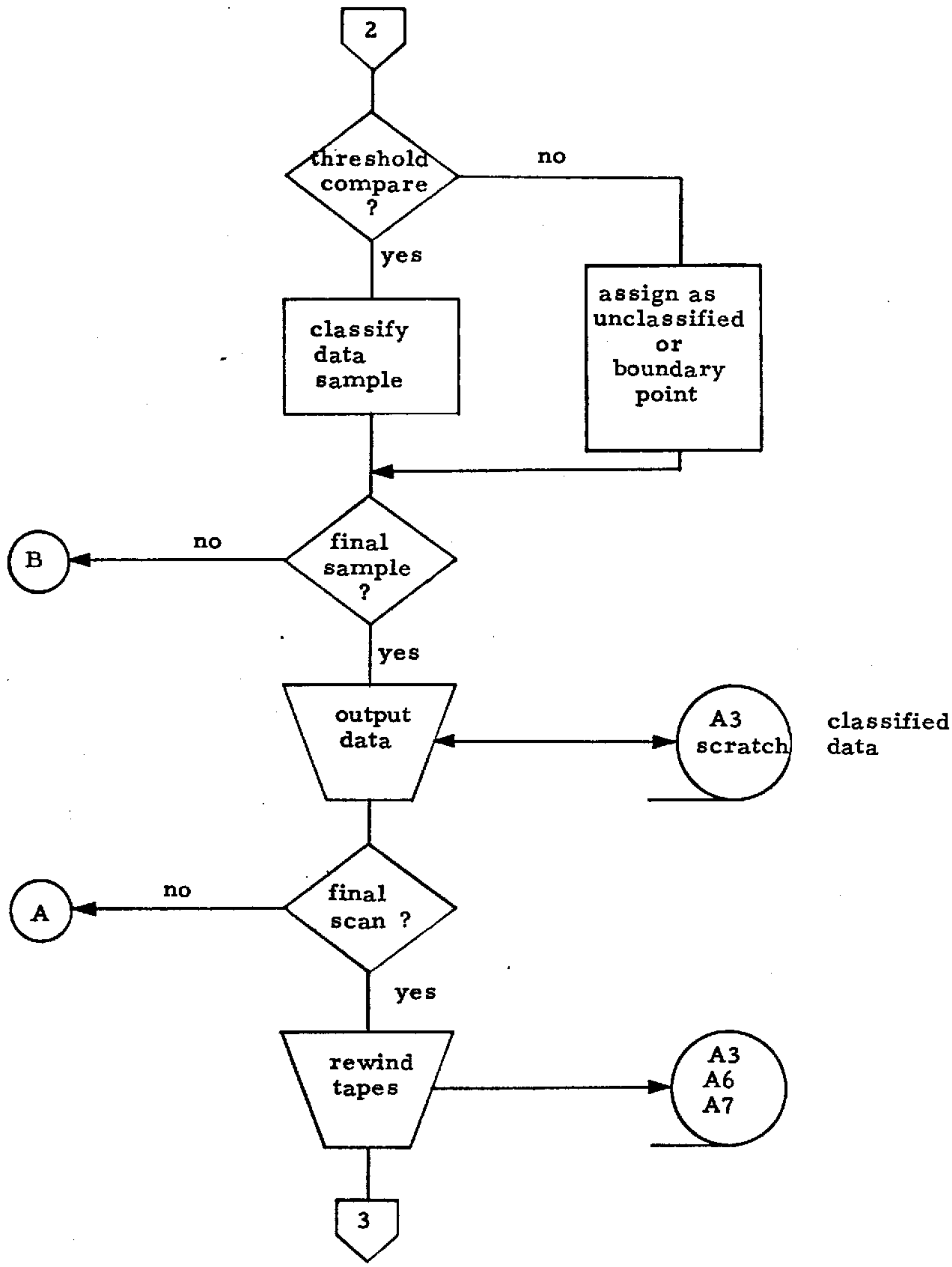


Spectral Discrimination

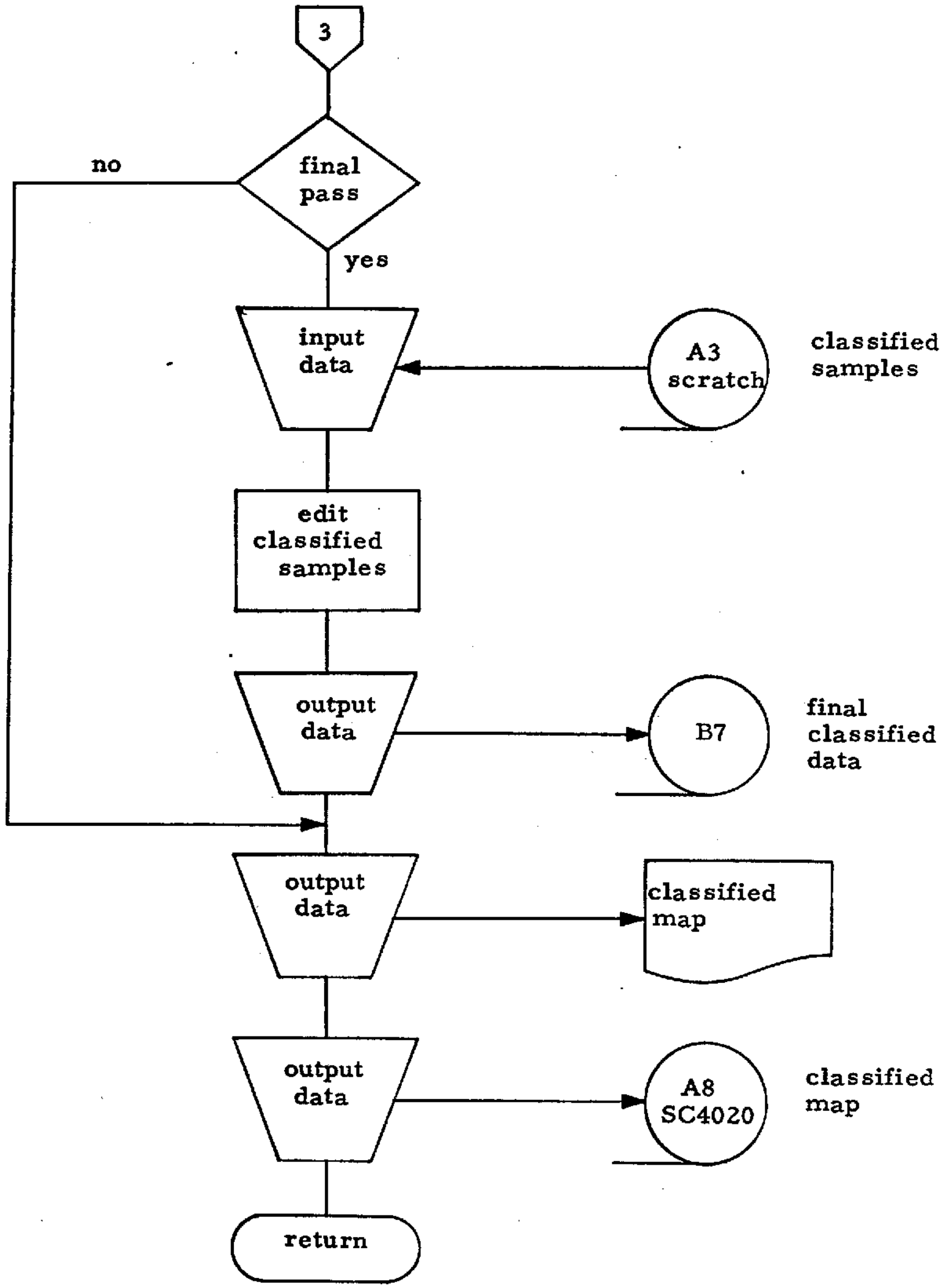

1
position tapes to NCNT + 1
A7
A6
NCNT = NCNT + 1
return



Spectral Discrimination



Spectral Discrimination



Spectral Discrimination
(concluded)

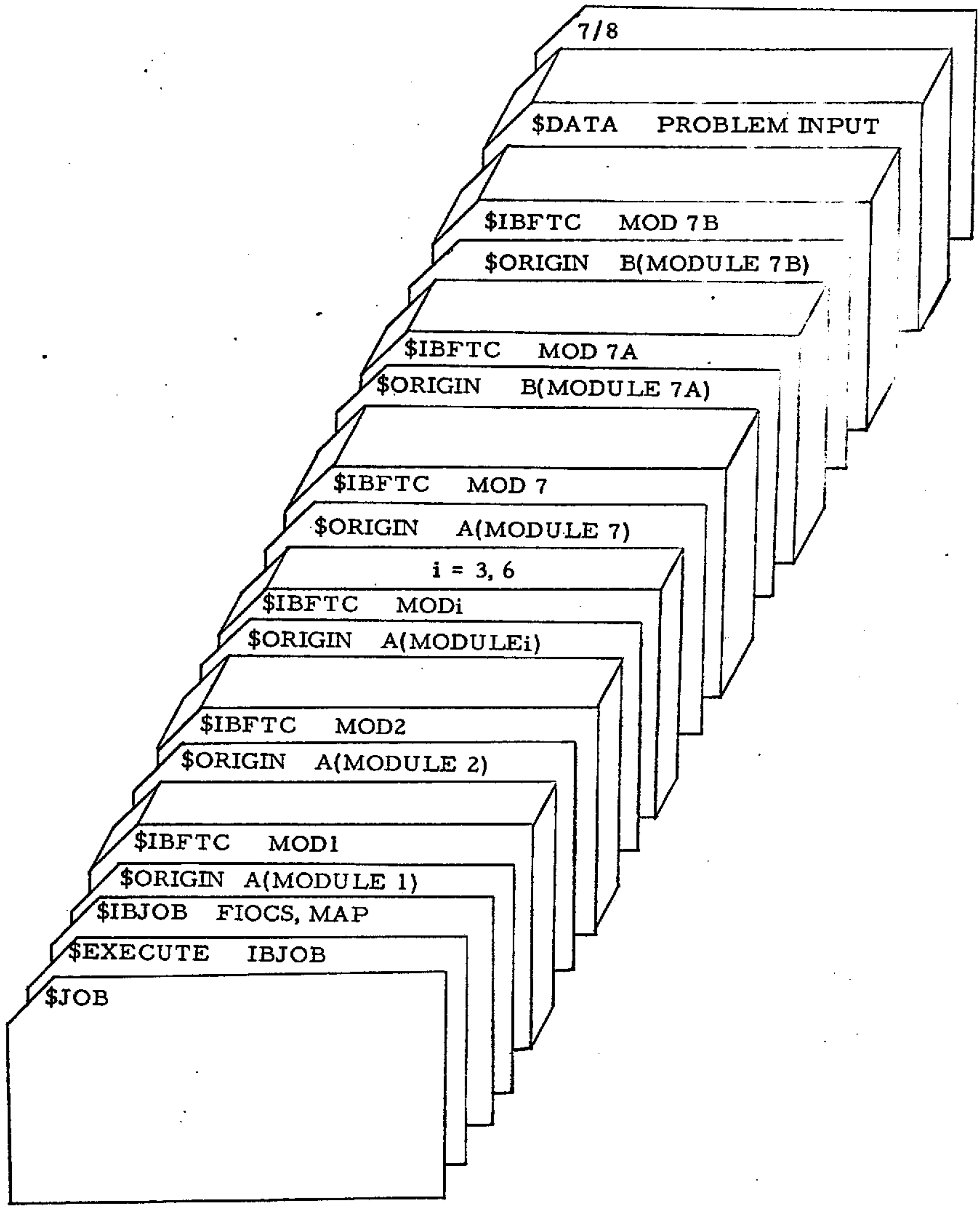


Figure 29 IBM 7094 Job Deck Setup

7094-__________ INSTRUCTIONS

NAME: John Doe	OP CODE: 11	STACK #___
BIN # 053 · LOC: 4663	JOB: # 123456	

IF EXCEEDS MAX: ☐ STR ☐ STZ ☐ DMP ☐ RETSY

☑ IBSYS ☐ SPOOK ☐ OTHER

☑ COMPL /ASSMBL ☑ EXECUTE ☐ PUNCH (BCD BIN)

☑ 4 FTRN ☐ 2 FTRN ☐ APT ☐ PERT

☐ MAP ☐ FAP ☐ SCAT ☐ OTHER

FAST TAPES: A B C D

INPUT TAPES			WORK
LOGIC	REEL NO	DEN	LOGIC
A6	56565	8	B6
			A7
			B7
			A3
			B5

LINES OF OUTPUT (1000'S) ☐ 0-5 ☑ 5-15 ☐ 15-30 ☐ OVER

MAXIMUM TIME: HOURS 3 MINUTES 0

NUMBER OF CASES _____

PROGRAMMER COMMENTS:

Comment, comment

OVER: _____

☐ SEE ON-LINE
☐ SEE TECHNIQUES
☐ MAX EXCEEDED
☐ RETURN TO SYS
☐ LINE MAX

OPERATOR COMMENTS:

OPER INIT: _____
OVER: _____

OUTPUT TAPES ONLY						4020
REEL NO.	LOGIC	DEN.	UNIT	NO OF CPYS	SAVE	TAPE
	B-1	8				
	B6	8			✓	
	B5	8			✓	
	B7	8			✓	
	A8	5				✓

NO FILES	NO FRAMES	COPIES		DENSITY		COPY-FLO		KALVAR
1	48	P	F	5	8	P	F	
						✓	✓	

MSFC - Form 533 (Rev February 1966)

Figure 30 IBM 7094 Instruction Form 533

APPENDIX

Computer Program Listing

*************I B M 7094 PROGRAM LISTING WITH JOB CARDS*************

```
$JOB           NASA JAYROE BIN 53 ,434300,00,12,I4MCE
$EXECUTE       IBJOB
$IBJOB         FIOCS,MAP
$FILE          -UNIT03-,NONE
$FILE          -UNIT04-,NONE
$FILE          -UNIT07-,NONE
$FILE          -UNIT08-,NONE
$FILE          -UNIT14-,NONE
$FILE          -UNIT15-,NONE
$IBFTC ERFDP
      DIMENSION MODULE(8)
      COMMON DATA(12)
      NAMELIST/INIT/MODULE
      DO 1 I=1,8
 1    MODULE(I)=0
      I=0
      READ(5,INIT)
      WRITE(6,INIT)
 998  CONTINUE
      I=I+1
      IGO=MODULE(I)
      IF (IGO .LE. 0) GO TO 999
      GO TO (10,11,12,13,14,15,16),IGO
 10   CONTINUE
      CALL PROB
      GO TO 998
 11   CONTINUE
      CALL GLEVEL
      GO TO 998
 12   CONTINUE
      CALL JONTPB
      GO TO 998
 13   CONTINUE
      CALL ISOMET
      GO TO 998
 14   CONTINUE
      CALL CNTLIN
      GO TO 998
 15   CONTINUE
      CALL BWNDR3
      GO TO 998
 16   CONTINUE
      CALL CLASFY
      GO TO 998
 999  CONTINUE
      CALL CLEAN
      STOP
      END

$ORIGIN        ALPHA,SYSUT2,REW
```

```
$IBFTC MOD1
      SUBROUTINE PROB
      COMMON DATA(12)
      COMMON/C1/ GLVL(11,12),NWHICH(12)
      DIMENSION NTABLE(201,12),DATAB(201)
      NAMELIST/INPUT1/NCH,NSPS,NSCANS,NSTART,NSTOP,NBTLG,MODE,
     1ITYPE,MSFC,LTN,NSKIP,
     1NCRE,XMAX,XMIN,NOCHS,NWHICH
      READ(5,INPUT1)
      WRITE(6,INPUT1)
C     CALCULATE PROBILITY DISTRIBUTION TABLE
      RESOL=(XMAX-XMIN)/200.0
      SCALE=200.0/(XMAX-XMIN)
C     INITILIZE ARRAYS
      DO 6 IB=1,12
      DO 6 IBB=1,201
      NTABLE(IBB,IB)=0
 6    CONTINUE
      DATAB(1)=XMIN
      DO 1 I=2,201
      DATAB(I)=DATAB(I-1)+RESOL
 1    CONTINUE
      NFLAG=0
      DO 4 IC=1,NSCANS
      NFLAG2=1
      DO 2 I=NSTART,NSTOP,NCRE
      CALL GET1(DATA,NSPS,0,NCH,NSCANO,LTN,IERR,NFLAG2,NFLAG,NSTART,
     1NBTLG,MODE,NCRE,ITYPE,MSFC)
      GO TO (10,999,10,10,999),IERR
 10   CONTINUE
      DO 3 IA=1,NOCHS
      NCHAN=NWHICH(IA)
      JJ=DATA(NCHAN)*SCALE+1.0
      IF (JJ .LT. 1) GO TO 3
      IF (JJ .GT. 201) GO TO 3
      NTABLE(JJ,NCHAN)=NTABLE(JJ,NCHAN)+1
 3    CONTINUE
 2    CONTINUE
 4    CONTINUE
 999  CONTINUE
      REWIND LTN
      WRITE(6,1000)
 1000 FORMAT(1H1,50X,26HPROBABILITY DISTRIBUTION                    )
 1001 FORMAT(1H0,130H AMPLITUDE    CH 1        CH 2        CH 3        CH 4
     1  CH 5        CH 6        CH 7        CH 8        CH 9        CH 10       CH 1
     21     CH 12     )
      WRITE(6,1001)
 1002 FORMAT(1H ,F9.1,12I10)
      DO 5 I=1,201
      WRITE(6,1002) DATAB(I),(NTABLE(I,JJ),JJ=1,NCH)
 5    CONTINUE
      CALL PRODEN(NTABLE,SCALE,GLVL,XMAX,XMIN)
 20   REWIND LTN
      RETURN
      END
```

```
$IBFTC PROBZ
      SUBROUTINE PRODEN (NTABLE,SCALE,GLVL,XMAX,XMIN    )
      DIMENSION NTABLE(201,12),GLVL(11,12)
      DO 3 IM=1,12
      NTOTAL=0
      MAX=201
      DO 1 I=1,MAX
      NTOTAL=NTOTAL+NTABLE(I,IM)
 1    CONTINUE
      NTEMP=0
      GLVL(1,IM)=0.0
      NCT=NTOTAL/10
      II=2
      DO 2 I=2,MAX
      IJ=I-1
      NTEMP=NTEMP+(NTABLE(IJ,IM)+NTABLE(I,IM))/2
      IF (NTEMP .LT. NCT) GO TO 2
      NTEMP=NTEMP-NCT
      GLVL(II,IM)=FLOAT(IJ)/SCALE+XMIN-.5
      II=II+1
 2    CONTINUE
      GLVL(11,IM)=XMAX
 3    CONTINUE
      DO 20 M=1,11
      WRITE(6,1000) (GLVL(M,MM),MM=1,12)
 20   CONTINUE
 1000 FORMAT(1X,12F6.1)
      RETURN
      END
```

```
$IBFTC MOD2
      SUBROUTINE GLEVEL
C
C              7094  DECK  -  MULTISPECTRAL SCANNER PLOT
C
      DIMENSION IALPHA(20)
      DIMENSION IARAY(256)
      DIMENSION IWAVE(13)
      DIMENSION NUM(20),PERCEN(20)
      DIMENSION XLIM(12)
      COMMON DATA(12)
      COMMON/C1/ GLVL(11,12),NWHICH(12)
      NAMELIST/INPUT2/NCH,NSPS,NSCANS,NSKIP,NSTART,NSTOP,ITERM,N,ICHAN,
     1IPRT,IPLT,INCX,INCY,NSTX,NSTY,NBTLG,MODE,NCRE,LTN,IOPT
      NAMELIST/GRYLVL/GLVL
      IN=5
      IN1=10
      IOUT=6
      NFLAG=0
```

```
      NFLGPT=0
      REWIND IN1
      CALL CAMRAV(35)
      CALL BUTTV(1)
      READ(5,INPUT2)
      WRITE(6,INPUT2)
      DO 303 I=1,N
      XLIM(I)=GLVL(I,ICHAN)
 303  CONTINUE
      READ (IN,GRYLVL)
      IF (IOPT .EQ. 0) GO TO 302
      DO 301 I=1,N
      XLIM(I)=GLVL(I,ICHAN)
 301  CONTINUE
 302  CONTINUE
C         ********************
C         ********************
      READ(IN,102)
      READ (IN,99) (IALPHA(I),I=1,N ),LLP
      READ(IN,109) XLOW,XUPP,(IWAVE(I),I=1,12)
      WRITE(IOUT,110)
 1003 FORMAT(1X,12A6,I6,5X,2F8.4)
  300 CONTINUE
      DO 777 I=1,N
      NUM(I)=0
777   CONTINUE
      KK=N-1
      WRITE(IOUT,106)
      WRITE(IOUT,110)
      WRITE(IOUT,102)
      WRITE(IOUT,106)
      WRITE(IOUT,103) ICHAN
      DO 2 I=1,KK
      IF(I.EQ.1) GO TO 9
      WRITE (6,105) IALPHA(I),XLIM(I),XLIM(I+1)
      GO TO 2
 9    WRITE(IOUT,112) (IWAVE(K),K=1, 12),IALPHA(I),XLIM(I),XLIM(I+1)
      GO TO 2
  2   CONTINUE
      WRITE(IOUT,106)
      CALL LABEL1(NSTART,NSTOP,NCRE)
C
C              INUM  COUNTS  SCAN  LINES
C
      INUM=0
      IDIF=NSTOP-NSTART+1
      IF (NSKIP .EQ. 0) GO TO 806
      DO 805 I=1,NSKIP
      NFLAG2=1
      CALL GET1(DATA,NSPS,0,NCH,NSCANO,LTN,IERR,NFLAG2,NFLAG,NSTART,
     1NBTLG,MODE,NCRE,ITYPE,MSFC)
 805  CONTINUE
 806  CONTINUE
      NFLAG1=0
   5  CONTINUE
```

```
      IIN=0
      NFLAG2=1
 205  DO 90 II=NSTART,NSTOP
      IIN=IIN+1
      CALL GET1(DATA,NSPS,    0,NCH,NSCANO,IN1,IERR,NFLAG2,NFLAG,NSTART,
     1NBTLG,MODE,NCRF,ITYPE,MSFC)
      GO TO(800,801,800,800,801),IERR
 801  WRITE(IOUT,100) NSCANO
      GO TO 999
 800  CONTINUE
      DO 601 I=2,N
      IF (DATA(ICHAN) .GT. XLIM(I)) GO TO 601
      NUM(I-1)=NUM(I-1)+1
      IARAY(IIN)=IALPHA(I-1)
      GO TO 90
 601  CONTINUE
 90   CONTINUE
      NUMTOT=INUM*IDIF
      TOTNUM=NUMTOT
      TOTPER=0.0
      DO 888 I=1,11
      XXXNUM=NUM(I)
      PERCEN(I)=(XXXNUM/TOTNUM) * 100.0
      TOTPER=TOTPER+PERCEN(I)
888   CONTINUE
  96  CONTINUE
      INUM=INUM+1
      IF (IPRT .EQ. 0) GO TO 889
      WRITE(IOUT,104) NSCANO,(IARAY(K),K=1,IDIF),LLP
 889  CONTINUE
      IF (IPLT .EQ. 0) GO TO 890
      CALL PLTBF1(IARAY,IDIF,NBLK,INCX,INCY,NSTX,NSTY,NCRE,
     1NFLGPT,NFLAG1)
 890  CONTINUE
      IF (NSCANO .GE. NSCANS) GO TO 218
      GO TO 5
  218 CONTINUE
      NFLAG1=0
      NFLGPT=0
      NSCANO=0
      NSTART=NSTOP+1
      NSTOP=NSTOP+IDIF
      IF (NSTOP .GE. NSPS) NSTOP=NSPS
      ITERM=ITERM-1
      REWIND IN1
      WRITE (IOUT,110)
      WRITE (IOUT,114)
      DO 666 I=1,11
      WRITE (IOUT,115) I,NUM(I),PERCEN(I),IALPHA(I)
666   CONTINUE
      WRITE (IOUT,116) NUMTOT,TOTPER
      IF(ITERM.GT.0)  GO TO 300
  98  WRITE(IOUT,110)
  99  FORMAT(25A1)
 100  FORMAT(16I4)
```

```
101   FORMAT(12F6.1)
 102  FORMAT(80H
    1                                                              )
103   FORMAT(2X,8HCHANNEL I2,36X,22HSYMBOL           INTERVAL /)
104   FORMAT(5X,I4,1X,1H*,120A1,A1)
105   FORMAT(50X,A1,8X,F6.1,3H - ,F6.1)
106   FORMAT(1X/)
108   FORMAT(10X,115A1)
109   FORMAT(2F10.0,12A1)
110   FORMAT(1H1/)
 111  FORMAT(2X,83HCROPS   C5-17,5-22,5-5,5-8,         ALTITUDE  2000 FT.
    1         DATE RECORDED  6/30/66 //)
112   FORMAT(2X,23HWAVE LENGTH (MICRONS)   ,12A1,13X,A1,8X,F6.1,
    13H - ,F6.1)
113   FORMAT(11X,112A1)
 114  FORMAT(20X,5HCLASS,7X,13HSAMPLE NUMBER,6X,7HPERCENT,10X,
    112HALPHA SYMBOL/)
115   FORMAT (21X,I2,10X,I6,12X,F7.2,15X,A1)
116   FORMAT (20X,5HTOTAL, 8X,I6,12X,F7.2)
117   FORMAT (3I4,12F7.3)
999   CONTINUE
      REWIND IN1
      RETURN
      END


$IBFTC PLOTT1
      SUBROUTINE LABEL1(NSTART,NSTOP,INCRE      )
      DIMENSION IOUT(120),JOUT(120),KOUT(120),LOUT(120)
      NDIF=(NSTOP-NSTART+1)/INCRE
      II=0
      DO 1 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/1000
      JOUT(II)=I/100-I/1000*10
      KOUT(II)=I/10-I/100*10
      LOUT(II)=I-I/10*10
      IF (LOUT(II) .LE. 0 ) LOUT(II)=0
1     CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      WRITE (6,10) (JOUT(I),I=1,NDIF)
      WRITE (6,10) (KOUT(I),I=1,NDIF)
      WRITE (6,10) (LOUT(I),I=1,NDIF)
10    FORMAT (11X,120I1)
      RETURN
      END


$IBFTC PLLTT1
      SUBROUTINE PLTBF1(ISYM,NN,NSQR,INCX,INCY,NSTX,NSTY,NCRF,
     1NFLAG,NFLAG1)
      DIMENSION ISYM(1),NBUFER(50)
      IF (NFLAG .NE. 0) GO TO 10
      NSQR=1024/IABS(INCY+INCX)
10    CONTINUE
```

```
      IF (NFLAG1 .NE. 0) GO TO 20
      CALL FRAMEV(0)
      NCOUNT=0
      INCRX=NSTX
      INCRY=NSTY
      NFLAG1=1
      IF (NFLAG .EQ. 0) GO TO 11
      CALL APRNTV (INCX,INCY,NN,NBUFER,INCRX,INCRY)
      INCRY=INCRY+INCX
      INCRX=INCRX+IABS(INCY)
 11   NFLAG=1
 20   CONTINUE
      NCOUNT=NCOUNT+1
      DO 1 I=1,50
      NBUFER(I)=0
 1    CONTINUE
      IA=0
      DO 2 I=1,NN
      IB=IA+6
      IC=IB/6
      ID=IB-IC*6
      IE=ID*6
      IA=IA+1
      CALL FLD(NBUFER(IC),IE,6, 0,ISYM(I))
 2    CONTINUE
      CALL APRNTV(INCX,INCY,NN,NBUFER,INCRX,INCRY)
      INCRY=INCRY+INCX
      INCRX=INCRX+IABS(INCY)
      IF (NCOUNT .GE. NSQR) NFLAG1=0
      RETURN
      END


$IBFTC GETDA1
      SUBROUTINE GET1(DATA,NSPS,NSKIP,NCPW,NSCANO,IRW,IERR,NFLAG2,
     1NFLAG,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC    )
      DIMENSION DATA(1),NDAT(890)
      DATA NFLAG/0/
      IF (NFLAG .NE. 0 ) GO TO 12
      NTEMP=NCPW*(NCRE-1)
      NSCANO=0
      NL=36-NBTLG
      NBLNG=36/NBTLG
      NFLAG=1
      NSTMP=(NSPS+NSTART-1)*NCPW+MSFC*19
      NSCANS=NSTMP
      NTMP=NSTMP/NBLNG
      NSTMP=NSTMP/NBLNG*NBLNG
      IF (NSTMP .LT. NSCANS) NTMP=NTMP+1
      NSCANS=NTMP
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      NSCANO=NSCANO+1
      CALL SKRBIN (IRW,1,RD   )
```

```
 401   CONTINUE
 402   CONTINUE
 12    CONTINUE
       IF (NFLAG2 .EQ. 0) GO TO 10
 13    CONTINUE
       M=(NSTART-1)*NCPW+MSFC*19
       IF (MODE .EQ. 1) GO TO 50
       CALL REDTPR (IRW,MODE,IERR,NW,NSCANS,NDAT)
       GO TO 51
 50    CALL REDTPC (IRW,MODE,IERR,NW,NSCANS,NDAT)
 51    CONTINUE
 1000  FORMAT (1X,I6)
       NNW=NW*6
       NFLAG2=0
       NSCANO=NSCANO+1
 10    CONTINUE
       DO 14 NN=1,NCPW
       IB=M+NBLNG
       ID=IB/NBLNG
       IF=IB-NBLNG*ID
       IBIT=NBTLG*IF
       M=M+1
       IF (ITYPE .EQ. 1) GO TO 15
       NDATA=0
       CALL FLD(NDATA,NL,NBTLG,IBIT,NDAT(ID))
       DATA(NN)=NDATA
       GO TO 14
 15    CONTINUE
       DATAN=0.0
       CALL FLD(DATAN,NL,NBTLG,IBIT,NDAT(ID))
       DATA(NN)=DATAN
 14    CONTINUE
       M=M+NTEMP
       RETURN
       END
```

```
$ORIGIN          ALPHA,SYSUT2,REW
$IBFTC MOD3
       SUBROUTINE JONTPB
       DIMENSION NKNT(4000),NP(4000,2),NWKCD( 750),NWKCL( 750)
       DIMENSION DATA(12)
C      DIMENSION IBIN( 750)
       DIMENSION ALPNUM(47),ALPHA(120),CORDX(3)
       DIMENSION IMX(6),IMY(6)
       NAMELIST /INPUT3/ NCH,NSPS,NSCANS,NSKIP,NSTART,NSTOP,LTN,NBTLG,
      1MODE,ITYPE,MSFC,NCRF,NOJP,IMX,IMY,SCALE,BIAS
       DATA ASTRIK/1H*/
       DATA XMARK/1HX/
       DATA BLANK/6H      /
 1060  FORMAT(48X,25HDATA SWITCH HAS OCCURRED      )
 1061  FORMAT(49X,30HJOINT PROBABILITY DISTRIBUTION    )
```

```
 1062 FORMAT(1H1)
 1063 FORMAT(44X,11H X-AXIS IS ,I6,6X,11H Y-AXIS IS ,I6)
 1040 FORMAT(1H ,67H MAXIMUM PROBABILITY OF UNCOMMONALITY EXCEEDED- CONT
     1INUE EXCUTION        ,I6)
 1064 FORMAT(1X,26HSYMBOL          N/SYMBOL       )
 1065 FORMAT(11X,121(1H*),/,11X,1H*,55X,5HPART ,I1,4H OF ,I1,53X,1H*,/
     1,11X,121(1H*))
      IRT=11
      NFLG3=0
      LIMIT=4000
      READ(5,INPUT3)
      WRITE(6,INPUT3)
      READ(5,1000) (ALPNUM(I),I=1,47)
 1000 FORMAT(1X,47A1)
C     REWIND IRT
      DO 80 NTIME=1,NOJP
      NFLG=0
      NH=1
      NV=2
      IF (NSKIP .EQ. 0) GO TO 9
      DO 8 I=1,NSKIP
      CALL SKRBIN(LTN,1,RD)
 8    CONTINUE
 9    CONTINUE
      MIX=IMX(NTIME)
      MIY=IMY(NTIME)
      DO 10 I=1,4000
 10   NKNT(I)=1
      WRITE (6,1062)
      WRITE(6,1061)
      WRITE(6,1063) MIX,MIY
      NCLS=1
      DO 13 MRP=1,NSCANS
      NFLAG2=1
      DO 15 MPR=1,NSPS
      IF (NFLG3 .GT. 0) GO TO 16
      CALL GET2(DATA,NSPS,0,NCH,NSCANO,LTN,IERR,NFLAG2,NSTART,
     1NBTLG,MODE,NCRF,ITYPE,MSFC)
      NC=DATA(MIX)*SCALE+BIAS
      NR=DATA(MIY)*SCALE+BIAS
      IF (NC .LT. 1) NC=1
      IF (NC .GT. 255) NC=255
      IF (NR .LT. 1) NR=1
      IF (NR .GT. 255) NR=255
      NP(1,NH)=NC
      NP(1,NV)=NR
      NFLG3=1
      GO TO 15
 16   CONTINUE
      CALL GET2(DATA,NSPS,0,NCH,NSCANO,LTN,IERR,NFLAG2,NSTART,
     1NBTLG,MODE,NCRF,ITYPE,MSFC)
      GO TO (996,999,998,996,999),IERR
 999  WRITE(6,1070) NSCANO
 1070 FORMAT(33HEND-OF-FILE ON INPUT AT SCAN NO. ,I6)
      GO TO 995
```

```
998   WRITE(6,1071)
1071  FORMAT(27HIRRECOVERABLE PARITY ERROR    )
      GO TO 995
996   CONTINUE
      NC=DATA(MIX)*SCALE+BIAS
      NR=DATA(MIY)*SCALE+BIAS
      IF (NC .LT. 1) NC=1
      IF (NC .GT. 255) NC=255
      IF (NR .LT. 1) NR=1
      IF (NR .GT. 255) NR=255
      DO 11 I=1,NCLS
      IF (NC .NE. NP(I,NH)) GO TO 11
      IF (NR .NE. NP(I,NV)) GO TO 11
      NKNT(I)=NKNT(I)+1
      GO TO 15
11    CONTINUE
      IF (NCLS .GE. LIMIT) GO TO 15
      NCLS=NCLS+1
      NP(NCLS,NH)=NC
      NP(NCLS,NV)=NR
15    CONTINUE
13    CONTINUE
17    CONTINUE
1020  FORMAT (1X,3I6)
C     FIND MINIMUM AND MAXIMUM IN ROW AND COLUMN DATA
      MINC=NP(1,NH)
      MAXC=NP(1,NH)
      MINR=NP(1,NV)
      MAXR=NP(1,NV)
      DO 25 I=1, NCLS
      IF (NP(I,NH) .GT. MAXC) MAXC=NP(I,NH)
      IF (NP(I,NV) .GT. MAXR) MAXR=NP(I,NV)
      IF (NP(I,NH) .LT. MINC) MINC=NP(I,NH)
      IF (NP(I,NV) .LT. MINR) MINR=NP(I,NV)
25    CONTINUE
C     TEST FOR SPREAD
      NDIFV=MAXR-MINR+1
      NDIFH=MAXC-MINC+1
      IF (NDIFV .GE. NDIFH) GO TO 19
      WRITE(6,1060)
      WRITE(6,1063) MIY,MIX
      NH=2
      NV=1
      NTEMP=MAXC
      MAXC=MAXR
      MAXR=NTEMP
      NTEMP=MINC
      MINC=MINR
      MINR=NTEMP
      NTEMP=NDIFH
      NDIFH=NDIFV
      NDIFV=NTEMP
19    CONTINUE
      WRITE(6,1064)
C     ORDER ROW DATA IN DECENDING ORDER
```

```
C     BUBBLE-UP
      NSORT=NCLS
 21   CONTINUE
      DO 20 I=2,NSORT
      IF (NP(I-1,NV) .GE. NP(I,NV)) GO TO 20
      NTEMP1=NP(I-1,NV)
      NTEMP2=NP(I-1,NH)
      NTEMP3=NKNT(I-1)
      NP(I-1,NV)=NP(I,NV)
      NP(I-1,NH)=NP(I,NH)
      NKNT(I-1)=NKNT(I)
      NP(I,NV)=NTEMP1
      NP(I,NH)=NTEMP2
      NKNT(I)=NTEMP3
 20   CONTINUE
      NSORT=NSORT-1
      IF (NSORT .GT. 1) GO TO 21
C     CALCULATE TABLE
      MAXKNT=0
      DO 22 I=1,NCLS
 22   IF (NKNT(I) .GT. MAXKNT) MAXKNT=NKNT(I)
      NFACT=MAXKNT/45
      XXX=FLOAT(MAXKNT)/46.0
      NFAC=0
      IF (NFACT .LT. 1) NFACT=1
      WRITE(6,1050) BLANK,NFAC
      NFAC=NFAC+NFACT
      DO 23 I=1,46
      WRITE(6,1050) ALPNUM(I),NFAC
 1050 FORMAT (3X,A6,6X,I6)
      NFAC=NFAC+NFACT
 23   CONTINUE
      GO TO 18
 14   CONTINUE
      WRITE(6,1040) LIMIT
      GO TO 17
 18   CONTINUE
C     PRINT DISTRIBUTION ON PAGE
C     SET UP LOOP
      ITERM=1
      IF (NDIFH .EQ. 120) GO TO 40
      ITERM=NDIFH/120+1
 40   CONTINUE
      NST=1
      NSO=NST+119
      IF (NSO .GT. NDIFH) NSO=NDIFH
      DO 50 LOOP=1,ITERM
      WRITE(6,1065) LOOP,ITERM
      CORDX(1)=FLOAT(NST+MINC-2)
      CORDX(2)=CORDX(1)+60.0
      CORDX(3)=CORDX(1)+110.0
 1021 FORMAT (1H1)
      IB=1
      DO 60 II=1,NDIFV
      DO 66 I=1,120
```

```
      ALPHA(I)=BLANK
 66   CONTINUE
C     IF (NFLG .EQ. 1) GO TO 69
C     DO 68 I=1,NDIFH
C     IBIN(I)=0
C68   CONTINUE
C69   CONTINUE
      IF (NP(IB,NV) .NE. MAXR-II+1) GO TO 65
      IA=0
      DO 26 I=IB,NCLS
      IE=I
      IF (NP(I,NV) .NE. NP(IB,NV)) GO TO 31
      IA=IA+1
      NWKCD(IA)=NP(I,NH)
      NWKCL(IA)=NKNT(I)
 26   CONTINUE
 31   CONTINUE
      IB=IE
C     ORDER X-AXIS
      NSORT=IA
      NUMCLX=IA
      IF (IA .LE. 1) GO TO 30
 28   CONTINUE
      DO 27 IC=2,NSORT
      IF (NWKCD(IC-1) .LE. NWKCD(IC)) GO TO 27
      NTEMP1=NWKCD(IC-1)
      NTEMP2=NWKCL(IC-1)
      NWKCD(IC-1)=NWKCD(IC)
      NWKCL(IC-1)=NWKCL(IC)
      NWKCD(IC)=NTEMP1
      NWKCL(IC)=NTEMP2
 27   CONTINUE
      NSORT=NSORT-1
      IF (NSORT .GE. 2) GO TO 28
 1030 FORMAT(1X,2I6)
 30   CONTINUE
C     IF (NFLG .EQ. 1) GO TO 91
C     IM=1
C     DO 90 I=1,NDIFH
C     IF (NWKCD(IM) .NE. 1) GO TO 90
C     IBIN(I)=NWKCL(IM)
C     IM=IM+1
C90   CONTINUE
C91   CONTINUE
      NM1=NST+MINC-1
      DO 67 ID=1,NUMCLX
      IF (NWKCD(ID) .LT. NM1) GO TO 67
      IC=ID
      GO TO 29
 67   CONTINUE
 29   CONTINUE
      IA=0
      DO 64 I=NST,NSO
      IA=IA+1
      IF (NWKCD(IC) .NE. MINC+I-1) GO TO 64
```

```
      XX=FLOAT(NWKCL(IC))/XXX
      ICAR=XX+(1.001-1.0/XXX)
      ALPHA(IA)=ALPNUM(ICAR)
      IC=IC+1
 64   CONTINUE
 65   CONTINUE
      IF (NFLG .EQ. 1) GO TO 70
C     WRITE(IRT) (IBIN(I),I=1,NDIFH)
 70   CONTINUE
      CORDY=FLOAT(MAXR-II+1)
      YMARG=XMARK
      IF (NDIFV-II .NE. (NDIFV-1-II/10*10)) YMARG=ASTRIK
      WRITE(6,1008) CORDY,YMARG,(ALPHA(I),I=1,120)
 1008 FORMAT(1X,F8.1,2X,A1,120A1)
 60   CONTINUE
      NST=NSO+1
      NSO=NST+119
      IF (NSO .GT. NDIFH) NSO=NDIFH
      WRITE(6,1010)
      WRITE(6,1011) CORDX(1),CORDX(2),CORDX(3)
 1011 FORMAT (6X,F10.4,50X,F10.4,40X,F10.4)
      NFLG=1
      WRITE(6,1062)
 50   CONTINUE
      REWIND LTN
C     END FILE IRT
 80   CONTINUE
C     REWIND IRT
 995  CONTINUE
 1010 FORMAT (11X,12(10HX*********))
      RETURN
      END


$IBFTC GETDA2
      SUBROUTINE GET2(DATA,NSPS,NSKIP,NCPW,NSCANO,IRW,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC    )
      DIMENSION DATA(1),NDAT(890)
      DATA NFLAG/0/
      IF (NFLAG .NE. 0 ) GO TO 12
      NTEMP=NCPW*(NCRE-1)
      NSCANO=0
      NL=36-NBTLG
      NBLNG=36/NBTLG
      NFLAG=1
      NSTMP=(NSPS+NSTART-1)*NCPW+MSFC*19
      NSCANS=NSTMP
      NTMP=NSTMP/NBLNG
      NSTMP=NSTMP/NBLNG*NBLNG
      IF (NSTMP .LT. NSCANS) NTMP=NTMP+1
      NSCANS=NTMP
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      NSCANO=NSCANO+1
      CALL SKRBIN (IRW,1,RD   )
```

```
 401   CONTINUE
 402   CONTINUE
 12    CONTINUE
       IF (NFLAG2 .EQ. 0) GO TO 10
 13    CONTINUE
       M=(NSTART-1)*NCPW+MSFC*19
       IF (MODE .EQ. 1) GO TO 50
       CALL REDTPR (IRW,MODE,IERR,NW,NSCANS,NDAT)
       GO TO 51
 50    CALL REDTPC (IRW,MODE,IERR,NW,NSCANS,NDAT)
 51    CONTINUE
 1000  FORMAT (1X,I6)
       NNW=NW*6
       NFLAG2=0
       NSCANO=NSCANO+1
 10    CONTINUE
       DO 14 NN=1,NCPW
       IB=M+NBLNG
       ID=IB/NBLNG
       IF=IB-NBLNG*ID
       IBIT=NBTLG*IF
       M=M+1
       IF (ITYPE .EQ. 1) GO TO 15
       NDATA=0
       CALL FLD(NDATA,NL,NBTLG,IBIT,NDAT(ID))
       DATA(NN)=NDATA
       GO TO 14
 15    CONTINUE
       DATAN=0.0
       CALL FLD(DATAN,NL,NBTLG,IBIT,NDAT(ID))
       DATA(NN)=DATAN
 14    CONTINUE
       M=M+NTEMP
       RETURN
       END



$ORIGIN         ALPHA,SYSUT2,REW
$IBFTC MOD4
       SUBROUTINE ISOMET
       DIMENSION NTA(256),NTB(256), DATA(256),TIME(256),DAT(12)
       DIMENSION NTC(256),NTD(256)
       NAMELIST/INPUT4/ NCH,NSPS,NSKIP,NBTLG,MODE,IRW,NCHAN,NSNCRE,NPCRE,
      1ITYPE,MSFC,
      2NPTSL,NPTSU,
      1MAXSCN,XMIN,XMAX,YMIN,YMAX,NBLSZX,NBLSZY,NSECT,NSMOV
      2,NDIREC
       READ (5,INPUT4)
       WRITE(6,INPUT4)
       NUP=512/NBLSZY
       NBLDIR=NBLSZX*NDIREC
       NTIMES=MAXSCN/NUP
       CALL CAMRAV(9)
       NBACK=1
```

```
      IF (NSKIP .EQ. 0) GO TO 33
      DO 32 I=1,NSKIP
      NFLAG2=1
      CALL GET3(DAT,NSPS,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,NBTLG,
     12    ,NPCRE,ITYPE,MSFC)
32    CONTINUE
33    CONTINUE
      DO 18 MMRR=1,NSECT
      NPTSUU=NPTSU
      XMIN=FLOAT(NPTSL)
      XMAX=FLOAT(NPTSUU)
      NDIFF=NPTSUU-NPTSL+1
      IF (NSMOV .NE. 0) NPTSUM=NDIFF -NSMOV+1
      NFLAG=0
      DO 16 MPR=1,NTIMES
      IF (NDIREC) 23,24,24
24    IXR=512-(512/NBLSZX-NUP)*NBLSZX
      IXL=0
      GO TO 25
23    IXL=512-(512/NBLSZX-NUP)*NBLSZX
      IXR=0
25    CONTINUE
      IYB=512
      IYT=0
      CALL FRAMEV(0)
      DO 10 MRP=1,NUP
      DO 17 MMRP=1,NSNCRE
      NFLAG2=1
      CALL GET3(DAT,NSPS,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,NBTLG,
     1MODE,NPCRE,ITYPE,MSFC)
      IF (NFLAG .EQ. 0) GO TO 19
17    CONTINUE
19    CONTINUE
      NFLAG=1
      DATA(1)=DAT(NCHAN)
      DO 1 I=2,NDIFF
      CALL GET3(DAT,NSPS,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,NBTLG,
     1MODE,NPCRE,ITYPE,MSFC)
      DATA(I)=DAT(NCHAN)
1     CONTINUE
      IF (NSMOV .EQ. 0) GO TO 22
      DO 20 I=1     ,NPTSUM
      IA=I
      IB=IA+NSMOV-1
      TEMP=0.0
      DO 21 II=IA,IB
      TEMP=TEMP+DATA(II)
21    CONTINUE
      TEMP=TEMP/FLOAT(NSMOV)
      DATA(I)=TEMP
20    CONTINUE
22    CONTINUE
      CALL SETMIV (IXL,IXR,IYB,IYT)
      CALL XSCALV (XMIN,XMAX,IXL,IXR)
      CALL YSCALV (YMIN,YMAX,IYB,IYT)
```

```
      DO 11 I=1,NDIFF
      TIME(I)=FLOAT(NPTSL-1+I)
      NTA(I)=NYV(DATA(I))
      NTB(I)=NXV(TIME(I))
 11   CONTINUE
      DO 12 I=2,NDIFF
      CALL LINEV (NTB(I-1),NTA(I-1),NTB(I   ),NTA(I   ))
 12   CONTINUE
      IF (MRP .LE. 1) GO TO 15
      DO 13 I=1,NDIFF
      CALL LINEV (NTD(I),NTC(I),NTB(I),NTA(I))
 13   CONTINUE
 15   CONTINUE
      DO 14 I=1,NDIFF
      NTD(I)=NTB(I)
      NTC(I)=NTA(I)
 14   CONTINUE
      IXL=IXL+NBLDIR
      IXR=IXR-NBLDIR
      IYB=IYB-NBLSZY
      IYT=IYT+NBLSZY
 10   CONTINUE
      CALL BSRECD(IRW,NBACK,NOF)
 16   CONTINUE
      REWIND IRW
      IF (NSKIP .EQ. 0) GO TO 30
      DO 31 I=1,NSKIP
      NFLAG2=1
      CALL GET3(DAT,NSPS,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,NBTLG,
     1MODE,NPCRE,ITYPE,MSFC)
 31   CONTINUE
 30   CONTINUE
      NPTSU=NPTSUU
      NTEMP=NPTSL
      NPTSL=NPTSU+1
      NPTSU=NPTSU+(NPTSU-NTEMP)+1
 18   CONTINUE
      RETURN
      END


$IBFTC GETDA3
      SUBROUTINE GET3(DATA,NSPS,NSKIP,NCPW,NSCANO,IRW,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC)
      DIMENSION DATA(1),NDAT(890)
      DATA NFLAG/0/
      IF (NFLAG .NE. 0 ) GO TO 12
      NTEMP=(NCRE-1)*NCPW
      NSCANO=0
      NL=36-NBTLG
      NBLNG=36/NBTLG
      NFLAG=1
      NSTMP=(NSPS+NSTART-1)*NCPW+MSFC*19
      NSCANS=NSTMP
      NTMP=NSTMP/NBLNG
```

```
      NSTMP=NSTMP/NBLNG*NBLNG
      IF (NSTMP .LT. NSCANS) NTMP=NTMP+1
      NSCANS=NTMP
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      NSCANO=NSCANO+1
      CALL SKRBIN (IRW,1,RD   )
 401  CONTINUE
 402  CONTINUE
 12   CONTINUE
      IF (NFLAG2 .EQ. 0) GO TO 10
 13   CONTINUE
      M=(NSTART-1)*NCPW+MSFC*19
      IF (MODE .EQ. 1) GO TO 50
      CALL REDTPR (IRW,MODE,IERR,NW,NSCANS,NDAT)
      GO TO 51
 50   CALL REDTPC (IRW,MODE,IERR,NW,NSCANS,NDAT)
 51   CONTINUE
 1000 FORMAT (1X,I6)
      NNW=NW*6
      NFLAG2=0
      NSCANO=NSCANO+1
 10   CONTINUE
      DO 14 NN=1,NCPW
      IB=M+NBLNG
      ID=IB/NBLNG
      IF=IB-NBLNG*ID
      IBIT=NBTLG*IF
      M=M+1
      IF (ITYPE .EQ. 1) GO TO 15
      NDATA=0
      CALL FLD(NDATA,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=NDATA
      GO TO 14
 15   CONTINUE
      DATAN=0.0
      CALL FLD(DATAN,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=DATAN
 14   CONTINUE
      M=M+NTEMP
      RETURN
      END
```

```
$ORIGIN        ALPHA,SYSUT2,REW
$IBFTC MOD5
      SUBROUTINE CNTLIN
      DIMENSION DAT(12)
      COMMON /LAB /NI3,NF3,NT3,Z(2560),KON(3400),NI2,NF2,NT2,U(3400),
     1V(3400)
```

```
      NAMELIST /INPUT5/ NCH,NSPS,IRW,NCHAN,NSNCRE,NPCRE,NPTSL,NPTSU,
     2NBTLG,MODE,
     2ITYPE,MSFC,
     3NSKIP,
     1MAXSCN,NSECT,  MSZX,  MSZY, BLK,FHINC,ZMIN,ZMAX,LAB
      READ (5,INPUT5)
      WRITE (6,INPUT5)
      CALL OPEN (IRW,1)
      IXR=1024.0-BLK+.5
      IXL=0
      IYB=0
      IYT=136
      NCRE=NPCRE
      NBLLK=1024.0/BLK+.5
      XNBLK= BLK
      FNPTSL=NPTSL
      FNPTSU=NPTSU
      NUP=MAXSCN/ ((MSZY-1)*NSNCRE)
      NSECT=NSPS/  MSZX
      NDIF=(NPTSU-NPTSL+1)/NPCRE
      NFLAG1=0
      FDIF=NDIF
      FRM=  MSZY
      CALL CAMRAV( 35)
      NWDS=NCH*NSPS
      CALL BUTTV(1)
      CALL SETMIV(IXL,IXR,IYB,IYT)
      IF (NSKIP .EQ. 0) GO TO 25
      DO 26 I=1,NSKIP
      NFLAG2=1
      CALL GET5(DAT,NSPS   ,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,
     1NBTLG,MODE,NPCRE,ITYPE,MSFC)
   26 CONTINUE
   25 CONTINUE
C     READ(IRW)
      III=0
      CALL FRAMEV(0)
      DO 18 MMRR=1,NSECT
      NPLTMP=NPTSL+1
      NPTSUU=NPTSU
      NPTSU=NPTSU-NPCRE
      NFLAG=0
      IMM=0
      DO 16 MRPP=1,NUP
      III=III+1
      IF (III .LE. NBLLK)GO TO 21
      CALL FRAMEV(0)
      III=1
      IXR=1024.0-BLK+.5
      IXL=0
   21 CONTINUE
      CALL YSCALV(1.0    ,FDIF   ,IYB,IYT)
      CALL XSCALV(1.0,FRM,IXL,IXR)
      DO 10 MRP=1,  MSZY
      IF (NFLAG1 .NE. 1) GO TO 20
```

```
      DO 22 IIM=1,NDIF
      IMM=IMM+1
      IMI=IMI+1
      Z(IMM)=Z(IMI)
 22   CONTINUE
      NFLAG1=0
      GO TO 10
 20   CONTINUE
      DO 17 MMRP=1,NSNCRE
      NFLAG2=1
 1001 FORMAT(1X,16F4.1)
      CALL GET5(DAT,NSPS  ,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,
     1NBTLG,MODE,NPCRE,ITYPE,MSFC)
      IF (NFLAG .EQ. 0) GO TO 19
 17   CONTINUE
 19   CONTINUE
      NFLAG=1
      IMM=IMM+1
      Z(IMM)=DAT(NCHAN)
      DO 1 I=NPLTMP,NPTSUU,NPCRE
      IMM=IMM+1
      CALL GET5(DAT,NSPS  ,0,NCH,NSCANO,IRW,IERR,NFLAG2,NPTSL,
     1NBTLG,MODE,NPCRE,ITYPE,MSFC)
      Z(IMM)=DAT(NCHAN)
 1    CONTINUE
 10   CONTINUE
 1000 FORMAT (1X,10F10.2)
      CALL CONTOR(MSZY,NDIF,FHINC,LAB,FNPTSL,FNPTSU,FRM,ZMIN,ZMAX)
      IXL=FLOAT(IXL)+BLK+.5
      IXR=FLOAT(IXR)-BLK+.5
      CALL SETMIV (IXL,IXR,IYB,IYT)
      IMM=0
      IMI=(MSZY-1)*NDIF
      NFLAG1=1
 16   CONTINUE
C     REWIND IRW
C     READ (IRW)
      NPTSU=NPTSUU
      NTEMP=NPTSL
      NPTSL=NPTSU-NPCRE
      NPTSU=NPTSU+(NPTSU-NTEMP)-NPCRE
 18   CONTINUE
      STOP
      END
$IBFTC GETDA5
      SUBROUTINE GET5(DATA,NSPS,NSKIP,NCPW,NSCANO,IRW,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC)
      DIMENSION DATA(1),NDAT( 890)
      DATA NFLAG/0/
      IF (NFLAG .NE. 0 ) GO TO 12
      NTEMP=(NCRE-1)*NCPW
      NSCANO=0
      NL=36-NBTLG
      NBLNG=36/NBTLG
      NFLAG=1
```

```
      NSTMP=(NSPS+NSTART-1)*NCPW+MSFC*19
      NSCANS=NSTMP
      NTMP=NSTMP/NBLNG
      NSTMP=NSTMP/NBLNG*NBLNG
      IF (NSTMP .LT. NSCANS) NTMP=NTMP+1
      NSCANS=NTMP
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      NSCANO=NSCANO+1
      CALL SKRBIN (IRW,1,RD   )
 401  CONTINUE
 402  CONTINUE
 12   CONTINUE
      IF (NFLAG2 .EQ. 0) GO TO 10
 13   CONTINUE
      M=(NSTART-1)*NCPW+MSFC*19
      IF (MODE .EQ. 1) GO TO 50
      CALL REDTPR (IRW,MODE,IERR,NW,NSCANS,NDAT)
      GO TO 51
 50   CALL REDTPC (IRW,MODE,IERR,NW,NSCANS,NDAT)
 51   CONTINUE
 1000 FORMAT (1X,I6)
      NNW=NW*6
      NFLAG2=0
      NSCANO=NSCANO+1
 10   CONTINUE
      DO 14 NN=1,NCPW
      IB=M+NBLNG
      ID=IB/NBLNG
      IF=IB-NBLNG*ID
      IBIT=NBTLG*IF
      M=M+1
      IF (ITYPE .EQ. 1) GO TO 15
      NDATA=0
      CALL FLD(NDATA,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=NDATA
      GO TO 14
 15   CONTINUE
      DATAN=0.0
      CALL FLD(DATAN,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=DATAN
 14   CONTINUE
      M=M+NTEMP
      RETURN
      END


$IBFTC CTOR
      SUBROUTINE CONTOR(L,M,FHINC,LAB,FNPTSL,FNPTSU,FRM,ZMIN,ZMAX)
      DIMENSION I(5),V1(3),V2(3),P1(3),P2(3),X(5),Y(5)
      DIMENSION V3(3)
      COMMON /LAB /NI3,NF3,NT3,Z(2560),KON(3400),NI2,NF2,NT2,U(3400),
     1V(3400)
      LOGICAL JCODE
      IF (M.GT.0.AND.L.GT.0.AND.FHINC.GT.0.0.AND.LAB.GE.0) GO TO 1
```

```
      WRITE (6,101) M,L,FHINC
      RETURN
 1    N=3400
      NC=(ZMAX-ZMIN)/ABS(FHINC)
      FH=ZMIN
 2    CONTINUE
      NLAB=LAB-1
      IF (LAB .EQ. 0) NLAB=LAB+1
      DO 15 IH=1,NC
      NI2=6
      NF2=5
      NT2=5
      V(2)=FNPTSU
      U(2)=FRM
      V(4)=FNPTSL
      U(4)=1.0
      K=1
      NLAB=NLAB+1
      IF (NLAB .NE. LAB) GO TO 51
      K=2
      NLAB=0
 51   DO 10 IL=2,L
      DO 10 IM=2,M
      I(3)=(IL-1)*M+IM
      I(2)=I(3)-1
      I(4)=I(3)-M
      I(1)=I(4)-1
      I(5)=I(1)
      X(1)=IM-1
      X(2)=IM-1
      X(3)=IM
      X(4)=IM
      X(5)=IM-1
      Y(1)=IL-1
      Y(2)=IL
      Y(3)=IL
      Y(4)=IL-1
      Y(5)=IL-1
      IZ2=I(2)
      IZ1=I(1)
      IZ3=I(3)
      IZ4=I(4)
      IZ5=I(5)
 1000 FORMAT (1X,2F10.2,I6)
      V1(1)=(X(1)+X(4))/2.0
      V1(2)=(Y(1)+Y(2))/2.0
      V1(3)=(Z(IZ1)+Z(IZ2)+Z(IZ3)+Z(IZ4))/4.0
      DO 4 J=1,4
      IZ=I(J)
      V2(1)=X(J)
      V2(2)=Y(J)
      V2(3)=Z(IZ)
      IZ=I(J+1)
      V3(1)=X(J+1)
      V3(2)=Y(J+1)
```

```
      V3(3)=Z(IZ)
      IF (V2(3) .LE. 0.0 .AND. V3(3) .LE. 0.0) GO TO 4
      NV2=V2(3)+.5
      NV3=V3(3)+.5
      IF (NV2 .EQ. NV3) GO TO 4
 1002 FORMAT(1X,3F10.2)
      CALL STRIKE(V1,V2,V3,FH,P1,P2,IER)
      IF (IER .NE. 2) GO TO 4
      IF (NF2+2 .LE. N) GO TO 59
      WRITE (6,100) FH,N
      GO TO 60
 59   NF2=NF2+2
      U(NF2-1)=P1(2)
      V(NF2-1)=P1(1)
      KON(NF2-1)=0
      U(NF2)=P2(2)
      V(NF2)=P2(1)
      KON(NF2)=1
 4    CONTINUE
 10   CONTINUE
 60   IF (NI2 .GE. NF2) GO TO 15
      JCODE=.TRUE.
      DO 61 J=NI2,NF2
      JTEMP=J
      DO 61 JJ=2,4,2
      IF (U(J) .EQ. U(JJ)) GO TO 68
      IF (V(J) .EQ. V(JJ)) GO TO 68
 61   CONTINUE
      NT2=NT2+2
      U(NT2-1)=U(NI2)
      V(NT2-1)=V(NI2)
      KON(NT2-1)=0
      U(NT2)=U(NI2+1)
      V(NT2)=V(NI2+1)
      KON(NI2)=1
      NI2=NI2+2
      GO TO 70
 68   J=JTEMP
      IF (KON(J) .EQ. 0) GO TO 62
      JJ=J-1
      JMAX=J
      GO TO 63
 62   JJ=J+1
      JMAX=JJ
 63   V1(1)=U(J)
      V1(2)=V(J)
      V2(1)=U(JJ)
      V2(2)=V(JJ)
      JDIF=JMAX-NI2-1
      DO 64 JSUB=1,JDIF
      JDEX=JMAX-JSUB+1
      KDEX=JMAX-JSUB-1
      U(JDEX)=U(KDEX)
 64   V(JDEX)=V(KDEX)
      NI2=NI2+2
```

```
      IF (JCODE) GO TO 72
      NT2=NT2+1
      U(NT2)=V2(1)
      V(NT2)=V2(2)
      KON(NT2)=1
      GO TO 70
 72   NT2=NT2+2
      U(NT2-1)=V1(1)
      V(NT2-1)=V1(2)
      KON(NT2-1)=0
      U(NT2)=V2(1)
      V(NT2)=V2(2)
      KON(NT2)=1
 70   IF (NI2 .GE. NF2) GO TO 16
      DO 71 J=NI2,NF2
      JTEMP=J
      IF (U(J).NE.U(NT2).OR.V(J).NE.V(NT2)) GO TO 71
      JCODE=.FALSE.
      GO TO 68
 71   CONTINUE
      GO TO 60
 16   CONTINUE
      NCNT=0
      JI=6
      DO 30 J=6,NT2
      JJ=J
      IF (KON(J) .EQ. 1) GO TO 31
      IF (NCNT .GT. 4) GO TO 33
      DO 32 JIJ=JI,JJ
      KON(JIJ)=0
 32   CONTINUE
 33   NCNT=0
      JI=J
      GO TO 30
 31   NCNT=NCNT+1
 30   CONTINUE
      IF (K .NE. 2) GO TO 801
      CALL LABEL(FH)
 801  CONTINUE
      DO 800 J=6,NT2
      JUL=NXV(U(J))
      JVL=NYV(V(J))
      JUR=NXV(U(J+1))
      JVR=NYV(V(J+1))
C     WRITE (6,1003) JUL,JVL,JUR,JVR
 1003 FORMAT(1X,4I6)
      IF (KON(J+1) .NE. 0) CALL LINEV(JUL,JVL,JUR,JVR)
 800  CONTINUE
  15  FH=FH+FHINC
 100  FORMAT (45H WARNING IN SUBROUTINE CONTOR AT A HEIGHT OF ,
     1F7.1,
     246HNUMBER OF POINTS EXCEEDED ALLOWABLE NUMBER OF ,
     3I5,/,
     421HEXECUTION CONTINUING ,//)
 101  FORMAT (28H ERROR IN SUBROUTINE CONTOR ,/I11,
```

```
     121H=NUMBER OF GRID ROWS ,/,I11,
     224H=NUMBER OF GRID COLUMNS ,/,F11.3,
     333H=DISTANCE BETWEEN CONTOUR LABELS ,/,I11,
     429H=FREQUENCY OF CONTOUR LABELS ,/,
     521HEXECUTION TERMINATED ,//  )
      RETURN
      END

$IBFTC STRK
      SUBROUTINE STRIKE(V1,V2,V3,FH,P1,P2,IER)
      DIMENSION V1(1),V2(1),V3(1),P1(1),P2(1),Q(3,3)
      DO 6 I=1,3
      P1(I)=0.0
      P2(I)=0.0
      Q(I,1)=V1(I)
      Q(I,2)=V2(I)
 6    Q(I,3)=V3(I)
 1000 FORMAT (1X,3F10.5)
      DO 7 I=1,3
      II=I
      DO 1 J=II,3
      IF (Q(3,II)-Q(3,J)) 1,1,2
 2    DO 8 L=1,3
      A=Q(L,II)
      Q(L,II)=Q(L,J)
      Q(L,J)=A
 8    CONTINUE
 1    CONTINUE
 7    CONTINUE
      IER=0
      IF(Q(3,3).LT.FH.OR. Q(3,1).GT.FH) RETURN
      IF(Q(3,1).EQ.FH.AND.Q(3,2).EQ.FH.AND.Q(3,3).EQ.FH) RETURN
      A=(FH-Q(3,1))/(Q(3,1)-Q(3,3))
      P1(1)=(Q(1,1)-Q(1,3))*A+Q(1,1)
      P1(2)=(Q(2,1)-Q(2,3))*A+Q(2,1)
      P1(3)=FH
      IF (Q(2,1) .NE. FH) GO TO 3
      IF (Q(3,2) .NE. FH) GO TO 4
      P2(1)=Q(1,2)
      P2(2)=Q(2,2)
      P2(3)=Q(3,2)
      IER=2
      RETURN
  3   IF (Q(3,2) ..GT. FH) GO TO 5
      A=(FH-Q(3,2))/(Q(3,2)-Q(3,3))
      P2(1)=(Q(1,2)-Q(1,3))*A+Q(1,2)
      P2(2)=(Q(2,2)-Q(2,3))*A+Q(2,2)
      P2(3)=FH
      IER=2
      RETURN
 4    IER=1
      P2(1)=P1(1)
      P2(2)=P1(2)
      P2(3)=FH
```

```
      RETURN
 5    A=(FH-Q(3,1))/(Q(3,1)-Q(3,2))
      P2(1)=(Q(1,1)-Q(1,2))*A+Q(1,1)
      P2(2)=(Q(2,1)-Q(2,2))*A+Q(2,1)
      P2(3)=FH
      IER=2
      RETURN
      END


$IBFTC MARK
      SUBROUTINE LABEL(H)
      COMMON /LAB /NI3,NF3,NT3,Z(2560),KON(3400),NI2,NF2,NT2,U(3400),
     1V(3400)
      I2=5
 12   I1=I2+1
      DIS=0.0
      IS=I1+1
      IF (I1+4 .GT. NT2) GO TO 99
      DO 10 I2=IS,NT2
      IF (KON(I2) .EQ. 0) GO TO 11
 10   DIS=DIS+SQRT((U(I2)-U(I2-1))**2+(V(I2)-V(I2-1))**2)
 11   II2=I2-1
      IF (DIS .LT. 270.0) GO TO 12
      IM=(II2+I1)/2
      IDIF=(II2-I1)/2
      IF (IDIF .LT. 2) GO TO 12
      DO 20 J=1,IDIF
      IL=IM-1
      IU=IM+J
      DIS=SQRT((U(IU)-U(IL))**2+(V(IU)-V(IL))**2)
      IF (DIS .GT. 68.0) GO TO 16
 20   CONTINUE
      GO TO 12
 16   IS=IL+1
      DO 19 JJ=IS,IU
 19   KON(JJ)=0
      REALM=10.0E+10
      IF (U(IU)-U(IL) .NE. 0.0) REALM=(V(IU)-V(IL))/(U(IU)-U(IL))
      ANGL=ATAN(REALM)
      R=(DIS-68.0)/2.0
      S=68.0+R
      IF (U(IU) .GT. U(IL)) GO TO 30
      II=IU
      IU=IL
      IL=II
 30   XP=(R*U(IU)+S*U(IL))/DIS+6.8*SIN(ANGL)
      YP=(R*V(IU)+S*V(IL))/DIS-6.8*COS(ANGL)
      DO 18 JJ=1,5
      ND=5-JJ
      IF (ABS(H) .LT. 10.0**(JJ-1)) GO TO 17
 18   CONTINUE
 17   CONTINUE
      IXP=XP
      IYP=YP
```

```
      CALL LABLV(H,IXP,IYP,ND,1,5)
      GO TO 12
 99   RETURN
      END
```

```
$ORIGIN         ALPHA,SYSUT2,REW
$IBFTC MOD6
      SUBROUTINE BWNDR3
      DIMENSION X(12,256),Y(12,256),MCHAN(12),NN(   256),KSYM(49),JSYM(
     1256)
      DIMENSION NWHICH(12)
      NAMELIST/INPUT6/NSCANS,NSTART,NSPS,NCH,NVAR,NSYM,ISUM,NBTLG,
     1MODE,ITYPE,MSFC ,NSKIP,NBLK,INCX,INCY,NSTX,NSTY,NCRE
      NAMELIST/NCHUSE/NWHICH
      EQUIVALENCE (NSCAN,NSCANS)
      EQUIVALENCE (NSTRT,NSTART)
      EQUIVALENCE (NCOL,NSPS)
      EQUIVALENCE (NCHAN,NCH)
      ICARD=5
      IPRINT=6
      INTAPE=10
      IOTAPE=11
      READ(ICARD,INPUT6)
      WRITE (IPRINT,INPUT6)
      READ(ICARD,NCHUSE)
      WRITE(IPRINT,NCHUSE)
    1 FORMAT(1X,7I4)
    3 FORMAT(1X,12I1)
      READ(ICARD,5)(KSYM(I),I=1,NSYM)
    5 FORMAT(1X,60A1)
      NFLAG=0
      AVE=ISUM
      APOP=0.0
      DXAVE=0.0
      DYAVE=0.0
      DZAVE=0.0
      NSAV=NSCAN
      IF (NSKIP .EQ. 0) GO TO 98
      DO 97 I=1,NSKIP
      CALL SKRBIN(INTAPE,1,NOP)
 97   CONTINUE
 98   CONTINUE
    2 FORMAT(1H1)
    4 FORMAT(5X,11I1)
      II=1
      KK=NSTRT-1
  160 IF(II.EQ.NSCAN) GO TO 510
      II=II+1
      NFLAG2=1
```

```
      KK=KK+1
      IF(II.NE.2) GO TO 290
      DO 170 JJ=1,NCOL
      CALL GET6(X(1,JJ),NCOL,0,NCHAN,NSCANO,INTAPE,IFRR,NFLAG2,NSTRT,
     1NBTLG,MODE,NCRF,ITYPE,MSFC)
  170 CONTINUE
  290 CONTINUE
      NFLAG2=1
      DO 300 JJ=1,NCOL
      CALL GET6(Y(1,JJ),NCOL,0,NCHAN,NSCANO,INTAPE,IFRR,NFLAG2,NSTRT,
     1NBTLG,MODE,NCRF,ITYPE,MSFC)
  300 CONTINUE
      DO 380 JJ=2,NCOL
      IJ=JJ-1
      XSUM=0.0
      YSUM=0.0
      ZSUM=0.0
      DO 360 ICHAN=1,ISUM
      IICHN=NWHICH(ICHAN)
      XDIFF=Y(IICHN,JJ)-Y(IICHN,IJ)
      YDIFF=Y(IICHN,JJ)-X(IICHN,JJ)
      XSUM=XSUM+XDIFF*XDIFF
      YSUM=YSUM+YDIFF*YDIFF
      ZSUM=ZSUM+XDIFF*YDIFF
  360 CONTINUE
      XSUM=XSUM/AVE
      YSUM=YSUM/AVE
      ZSUM=ZSUM/AVE
      APOP=APOP+1.0
      AA=1.0/APOP
      BB=1.0-AA
      DXAVE=BB*DXAVE+AA*YSUM
      DYAVE=BB*DYAVE+AA*XSUM
      DZAVE=BB*DZAVE+AA*ZSUM
      XSUM=SQRT(XSUM)
      IF (XSUM .LT. 53.4) GO TO 365
 364  XSUM=53.0
      YSUM=53.0
      GO TO 366
 365  CONTINUE
      YSUM=SQRT(YSUM)
      IF (YSUM .LT. 53.4) GO TO 366
      GO TO 364
 366  CONTINUE
      CALL JNTPB(YSUM,XSUM,NFLAG,0,0,KSYM(3),NPOP,
     2DXAVE,DYAVE,
     3DZAVE,
     1JJ,NSCAN,II,NCOL,JSYM,X,NSTRT,NN,INCX)
      NSCAN=NSAV
  380 CONTINUE
      DO 500 JJ=1,NCOL
      DO 490 ICHAN=1,ISUM
      IICHN=NWHICH(ICHAN)
      X(IICHN,JJ)=Y(IICHN,JJ)
  490 CONTINUE
```

```
  500 CONTINUE
      GO TO 160
 510  CONTINUE
      NFLAG=1
      CALL JNTPB(YSUM,XSUM,NFLAG,0,0,KSYM(3),NPOP,
     2DXAVE,DYAVE,
     3DZAVE,
     1JJ,NSCAN,II,NCOL,JSYM,X,NSTRT,NN,INCX)
      REWIND IOTAPE
      REWIND INTAPE
C     CALL CLEAN
      RETURN
      END


$IBFTC PLOTT6
      SUBROUTINE LABEL6(NSTART,NSTOP,INCRE      )
      DIMENSION IOUT(120)
      NDIF=(NSTOP-NSTART+1)/INCRE
      II=0
      DO 1 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/1000
 1    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 2 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/100-I/1000*10
 2    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 3 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/10-I/100*10
 3    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 4 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I-I/10*10
      IF (IOUT(II) .LE. 0 ) IOUT(II)=0
 4    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
 10   FORMAT (11X,120I1)
      RETURN
      END


$IBFTC GETDA6
      SUBROUTINE GET6(DATA,NSPS,NSKIP,NCPW,NSCANO,IRW,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSEC    )
      DIMENSION DATA(1),NDAT( 890)
      DATA NFLAG/0/
      IF (NFLAG .NE. 0 ) GO TO 12
```

```
      NSCANO=0
      NTEMP=(NCRE-1)*NCPW
      NL=36-NBTLG
      NBLNG=36/NBTLG
      NFLAG=1
      NSTMP=(NSPS+NSTART-1)*NCPW+MSFC*19
      NSCANS=NSTMP
      NTMP=NSTMP/NBLNG
      NSTMP=NSTMP/NBLNG*NBLNG
      IF (NSTMP .LT. NSCANS) NTMP=NTMP+1
      NSCANS=NTMP
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      NSCANO=NSCANO+1
      CALL SKRBIN (IRW,1,RD   )
 401  CONTINUE
 402  CONTINUE
 12   CONTINUE
      IF (NFLAG2 .EQ. 0) GO TO 10
 13   CONTINUE
      M=(NSTART-1)*NCPW+MSFC*19
      IF (MODE .EQ. 1) GO TO 50
      CALL REDTPR (IRW,MODE,IERR,NW,NSCANS,NDAT)
      GO TO 51
 50   CALL REDTPC (IRW,MODE,IERR,NW,NSCANS,NDAT)
 51   CONTINUE
 1000 FORMAT (1X,I6)
      NNW=NW*6
      NFLAG2=0
      NSCANO=NSCANO+1
 10   CONTINUE
      DO 14 NN=1,NCPW
      IB=M+NBLNG
      ID=IB/NBLNG
      IF=IB-NBLNG*ID
      IBIT=NBTLG*IF
      M=M+1
      IF (ITYPE .EQ. 1) GO TO 15
      NDATA=0
      CALL FLD(NDATA,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=NDATA
      GO TO 14
 15   CONTINUE
      DATAN=0.0
      CALL FLD(DATAN,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=DATAN
 14   CONTINUE
      M=M+NTEMP
      RETURN
      END


$IBFTC JOINT
      SUBROUTINE JNTPB(DATAH,DATAV,NFLAG,MIX,MIY,ALPNUM,NPOP,
     2DXAVE,DYAVE,
```

```
     3DZAVE,
     1JJ,NSCAN,ISCAN,NCOL,ISYM,NX,NSTRT,NN,INCXY)
      DIMENSION INCXY(1)
      DIMENSION NP(54,54)
      DIMENSION DATA(12)
      DIMENSION IBIN(255 )
      DIMENSION ISYM(1),IQUIT(100)
      DIMENSION NX(1),NN(1)
      DIMENSION ALPNUM( 1),ALPHA(120),CORDX(3)
      DOUBLE PRECISION A(2,2),EIGEN(2,2)
      INTEGER ALPNUM,ALPHA,BLANK
      DATA ASTRIK/1H*/
      DATA XMARK/1HX/
      DATA BLANK/6H      /
      DATA NFLAG4/0/
 1060 FORMAT(48X,25HDATA SWITCH HAS OCCURRED     )
 1061 FORMAT(49X,30HJOINT PROBABILITY DISTRIBUTION  )
 1062 FORMAT(1H1)
 1063 FORMAT(44X,11H X-AXIS IS ,I6,6X,11H Y-AXIS IS ,2I6)
 1066 FORMAT (30X,6HDXAVE=,F15.7,6HDYAVE=,F15.7,6HDZAVE=,F15.7     )
 1040 FORMAT(1H ,67H MAXIMUM PROBABILITY OF UNCOMMONALITY EXCEEDED- CONT
     1INUE EXCUTION          ,2I6)
 1064 FORMAT(1X,26HSYMBOL             N/SYMBOL       )
 1065 FORMAT(11X,121(1H*),/,11X,1H*,55X,5HPART ,I1,4H OF ,I1,53X,1H*,/
     1,11X,121(1H*))
      IF (NFLAG4 .GT. 0) GO TO 80
      NFLG3=0
 1000 FORMAT(1X,47A1)
      NFLG=0
      NI=2
      IRW=1
      NFLGEN=0
      IRT=11
      DO 1 I=1,54
      DO 1 J=1,54
      NP(I,J) =0
1     CONTINUE
      REWIND IRT
      REWIND IRW
      NFLAG4=1
80    CONTINUE
      IF (NFLAG.GT. 0) GO TO 13
      NC=DATAV+1.5
      NR=DATAH+1.5
      IF (NC .LT. 1) NC=1
      IF (NR .LT. 1) NR=1
      NP(NR,NC)=NP(NR,NC)+1
      I=54*(NC-1)+NR
      IBIN(JJ)=I
      IF (JJ .LT. NCOL) GO TO 15
      IBIN(1)=IBIN(2)
      WRITE(IRW)(IBIN(II),II=1,NCOL)
15    CONTINUE
      RETURN
13    CONTINUE
```

```
 17   CONTINUE
      REWIND IRW
      IOPT=1
      IN=2
      IM=2
      RHO=1.0/(10.0**5)
      A(1,1)=DXAVE
      A(2,2)=DYAVE
      A(1,2)=DZAVE
      A(2,1)=DZAVE
      CALL DJCOBI(A,IM,IN,IOPT,RHO,ERR,EIGEN)
C     WRITE(6,1067) A(1,1),A(1,2),EIGEN(1,1),EIGEN(1,2)
C     WRITE(6,1067) A(2,1),A(2,2),EIGEN(2,1),EIGEN(2,2)
 1067 FORMAT(1X,2E15.7,10X,2E15.7)
      DXAVE=1.0/A(1,1)
      DYAVE=1.0/A(2,2)
      A(1,1)=EIGEN(1,1)*DXAVE
      A(1,2)=EIGEN(2,1)*DXAVE
      A(2,1)=EIGEN(1,2)*DYAVE
      A(2,2)=EIGEN(2,2)*DYAVE
      DXAVE=EIGEN(1,1)*A(1,1)+EIGEN(1,2)*A(2,1)
      DYAVE=EIGEN(2,1)*A(1,2)+EIGEN(2,2)*A(2,2)
      DZAVE=EIGEN(1,1)*A(1,2)+EIGEN(1,2)*A(2,2)+EIGEN(2,1)*A(1,1)
     1+EIGEN(2,2)*A(2,1)
      WRITE(6,1066) DXAVE,DYAVE,DZAVE
      I=0
      DO 130 NC=1,54
      DO 130 NR=1,54
      I=I+1
      IF (NP(NR,NC) .EQ. 0) GO TO 130
      XXX=NR*NR
      YYY=NC*NC
      ZZZ=NR*NC
      SUM=DXAVE*XXX+DYAVE*YYY
     1+DZAVE*ZZZ
      IF(SUM.GE.1.0)GO TO 115
      NX(I)=0
      GO TO 130
 115  NX(I)=-1
  130 CONTINUE
      WRITE (6,1062)
      WRITE(6,1061)
      WRITE(6,1066) DXAVE,DYAVE,DZAVE
      WRITE(6,1064)
C     CALCULATE TABLE
      MAXKNT=0
      DO 22 NC=1,53
      DO 22 NR=1,53
      IF (NP(NR,NC) .GT. MAXKNT) MAXKNT=NP(NR,NC)
 22   CONTINUE
      IF (MAXKNT .LT. 46 ) MAXKNT=46
      NFACT=MAXKNT/45
      XXX=FLOAT(MAXKNT)/46.0
      NFAC=0
      IF (NFACT .LT. 1) NFACT=1
```

```
      WRITE(6,1050) BLANK,NFAC
      NFAC=NFAC+NFACT
      DO 23 I=1,46
      WRITE(6,1050) ALPNUM(I),NFAC
 1050 FORMAT (3X,A6,6X,I6)
      NFAC=NFAC+NFACT
 23   CONTINUE
      WRITE(6,1062)
C     PRINT DISTRIBUTION ON PAGE
      CORDX(1)=0.0
      CORDX(2)=CORDX(1)+60.0
      CORDX(3)=CORDX(1)+110.0
 1021 FORMAT (1H1)
      DO 65 IEND=1,54
      NC=55-IEND
      DO 66 I=1,54
      ALPHA(I)=BLANK
 66   CONTINUE
      IF (NFLG .EQ. 1) GO TO 69
      DO 68 I=1,54
      IBIN(I)=0
 68   CONTINUE
 69   CONTINUE
      DO 64 NR=1,54
      XX=FLOAT(NP(NR,NC))
      IF (NFLG .EQ. 1) GO TO 91
      IBIN(NR)=NP(NR,NC)
 91   CONTINUE
      ICAR=XX+(1.001-1.0/XXX)
      IF (ICAR .GT. 46 ) ICAR=46
      ALPHA(NR)=ALPNUM(ICAR)
 64   CONTINUE
      CORDY=FLOAT(NC)
      YMARG=XMARK
      IF (NC .NE. 54-IEND/10*10) YMARG=ASTRIK
      WRITE(6,1008) CORDY,YMARG,(ALPHA(I),I=1,54)
 1008 FORMAT(1X,F8.1,2X,A1,120A1)
 65   CONTINUE
      WRITE(6,1010)
      WRITE(6,1011) CORDX(1),CORDX(2),CORDX(3)
 1011 FORMAT (6X,F10.4,50X,F10.4,40X,F10.4)
      NFLG=1
      WRITE(6,1062)
      NSUB=1
      LWFR=1
      LOW=NSTRT
 705  CONTINUE
      NHI=LOW+120-1
      NUPPER=LWFR+120-1
      IF (NUPPER .GT. NCOL) NUPPER=NCOL
      WRITE(6,1062)
      CALL LABEL6(LOW,NHI,1)
      DO 131 II=NI,NSCAN
      READ(IRW) (IBIN(JJ),JJ=1,NCOL)
      DO 135 JJ=1,NCOL
```

```
      ICHECK=IBIN(JJ)
      JCHECK=NX(ICHECK)
      IF (JCHECK .NE. 0) GO TO 117
      ISYM(JJ)=ALPNUM(NSUB-1)
      NN(JJ)=0
      GO TO 135
 117  ISYM(JJ)=ALPNUM(NSUB)
      NN(JJ)=-1
 135  CONTINUE
      IF (NFLGEN .NE. 0) GO TO 136
      WRITE(IRT) (NN(JJ),JJ=1,NCOL)
      CALL PLTBF6(ISYM,NCOL,NBLK,INCXY(1),INCXY(2),INCXY(3),
     1INCXY(4),NCRE)
 136  CONTINUE
      WRITE(6,1036) II,(ISYM(JJ),JJ=LWER,NUPPER)
 1036 FORMAT(5X,I6,120A1)
 1035 FORMAT(1X,I6)
 131  CONTINUE
      NFLGEN=1
      REWIND IRW
      LWER=NUPPER+1
      LOW=NHI+1
      IF  (NUPPER .LT. NCOL) GO TO 705
 995  CONTINUE
 1010 FORMAT (11X,12(10HX*********))
      NFLGEN=0
      RETURN
      END


$IBFTC PLLT6
      SUBROUTINE PLTBF6(ISYM,NN,NSQR,INCX,INCY,NSTX,NSTY,NCRE)
      DIMENSION ISYM(1),NBUFER(50)
      DATA NFLAG/0/
      DATA NFLAG1/0/
      IF (NFLAG .NE. 0) GO TO 10
      CALL CAMRAV(35)
      CALL BUTTV(1)
 10   CONTINUE
      IF (NFLAG1 .NE. 0) GO TO 20
      CALL FRAMEV(0)
      NCOUNT=0
      INCRX=NSTX
      INCRY=NSTY
      NFLAG1=1
      IF (NFLAG .EQ. 0) GO TO 11
      CALL APRNTV (INCX,INCY,NN,NBUFER,INCRX,INCRY)
      INCRY=INCRY+INCX
      INCRX=INCRX+IABS(INCY)
      NCOUNT=NCOUNT+1
 11   NFLAG=1
 20   CONTINUE
      NCOUNT=NCOUNT+1
      DO 1 I=1,50
      NBUFER(I)=0
```

```
 1    CONTINUE
      IA=0
      DO 2 I=1,NN
      IB=IA+6
      IC=IB/6
      ID=IB-IC*6
      IE=ID*6
      IA=IA+1
      CALL FLD(NBUFFER(IC),IE,6, 0,ISYM(I))
 2    CONTINUE
      CALL APRNTV(INCX,INCY,NN,NBUFFER,INCRX,INCRY)
      INCRY=INCRY+INCX
      INCRX=INCRX+IABS(INCY)
      IF (INCRX .GE. 1023) NFLAG1=0
      RETURN
      END
```

```
$ORIGIN         ALPHA,SYSUT2,REW
$IBFTC MOD7
      SUBROUTINE CLASFY
      COMMON /LAB1/XBAR(42,12),SIGMA(42,12),ROT(42,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IXXX,IYYY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSFC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      NAMELIST/INPUT7/NPASS,NCLUST
      NAMELIST/INPUTA/NSPS,NSCANS,NCH,  LT1,LT9,LT10,LT11,LT12,LT13,
     1NSTART,NSTOP,NBTLG,MODE,ITYPE,MSFC,I4,NCRE,NSKIP,INCX,INCY,NSTX,
     2NSTY,IXXX,IYYY
      EQUIVALENCE (NCH,NCHAN)
      READ(5,INPUT7)
      READ(5,INPUTA)
      WRITE(6,INPUTA)
      READ(5,1006) (ALPHA(I),I=1,48)
 1006 FORMAT(1X,60A1)
      KOUNT=NCLUST
      NSCANS=NSCANS-1
      INITCL=NCLUST+1
      DO 1 I=1,NPASS
      CALL TRUCK(NCLUST,NPASS    )
      CALL SEQMRG (NCLUST,KOUNT,INITCL      )
      CALL CLASS (KOUNT , I, NPASS     )
      NCLUST=KOUNT
      INITCL=KOUNT+1
 1    CONTINUE
      RETURN
      END
```

```
$IBFTC GETDA7
      SUBROUTINE GET7(DATA,NSPS,NSKIP,NCPW,NSCANO,IRW,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC    )
      DIMENSION DATA(1),NDAT( 890)
      DATA NFLAG/0/
      IF (NFLAG .NE. 0 ) GO TO 12
      NSCANO=0
      NTEMP=NCPW*(NCRE-1)
      NL=36-NBTLG
      NBLNG=36/NBTLG
      NFLAG=1
      NSTMP=(NSPS+NSTART-1)*NCPW+MSFC*19
      NSCANS=NSTMP
      NTMP=NSTMP/NBLNG
      NSTMP=NSTMP/NBLNG*NBLNG
      IF (NSTMP .LT. NSCANS) NTMP=NTMP+1
      NSCANS=NTMP
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      NSCANO=NSCANO+1
      CALL SKRBIN (IRW,1,RD   )
 401  CONTINUE
 402  CONTINUE
 12   CONTINUE
      IF (NFLAG2 .EQ. 0) GO TO 10
 13   CONTINUE
      M=(NSTART-1)*NCPW+MSFC*19
      IF (MODE .EQ. 1) GO TO 50
      CALL REDTPR (IRW,MODE,IERR,NW,NSCANS,NDAT)
      GO TO 51
 50   CALL REDTPC (IRW,MODE,IERR,NW,NSCANS,NDAT)
 51   CONTINUE
 1000 FORMAT (1X,I6)
      NNW=NW*6
      NFLAG2=0
      NSCANO=NSCANO+1
 10   CONTINUE
      DO 14 NN=1,NCPW
      IB=M+NBLNG
      ID=IB/NBLNG
      IF=IB-NBLNG*ID
      IBIT=NBTLG*IF
      M=M+1
      IF (ITYPE .EQ. 1) GO TO 15
      NDATA=0
      CALL FLD(NDATA,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=NDATA
      GO TO 14
 15   CONTINUE
      DATAN=0.0
      CALL FLD(DATAN,NL,NBTLG,IBIT,NDAT(ID))
      DATA(NN)=DATAN
 14   CONTINUE
      M=M+NTEMP
```

```
      RETURN
      END

$IBFTC PLOTT7
      SUBROUTINE LABEL7 (NSTART,NSTOP,INCRE      )
      DIMENSION IOUT(120)
      NDIF=(NSTOP-NSTART+1)/INCRE
      II=0
      DO 1 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/1000
 1    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 2 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/100-I/1000*10
 2    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 3 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I/10-I/100*10
 3    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
      II=0
      DO 4 I=NSTART,NSTOP,INCRE
      II=II+1
      IOUT(II)=I-I/10*10
      IF (IOUT(II) .LE. 0 ) IOUT(II)=0
 4    CONTINUE
      WRITE (6,10) (IOUT(I),I=1,NDIF)
 10   FORMAT (11X,120I1)
      RETURN
      END

$IBFTC PLLTT7
      SUBROUTINE PLTBF7(ISYM,NN,NSQR,INCX,INCY,NSTX,NSTY,NCRE,
     1NFLAG,NFLAG1)
      DIMENSION ISYM(1),NBUFER(50)
      DATA NFLG/0/
      IF (NFLG  .NE. 0) GO TO 10
      NFLG=1
      CALL CAMRAV(35)
      CALL BUTTV(1)
      NSQR=1024/IABS(INCY+INCX)
 10   CONTINUE
      IF (NFLAG1 .NE. 0) GO TO 20
      CALL FRAMEV(0)
      NCOUNT=0
      INCRX=NSTX
      INCRY=NSTY
      NFLAG1=1
```

```
      IF (NFLAG .EQ. 0) GO TO 11
      CALL APRNTV (INCX,INCY,NN,NBUFER,INCRX,INCRY)
      INCRY=INCRY+INCX
      INCRX=INCRX+IABS(INCY)
      NCOUNT=NCOUNT+1
 11   NFLAG=1
 20   CONTINUE
      NCOUNT=NCOUNT+1
      DO 1 I=1,50
      NBUFER(I)=0
 1    CONTINUE
      IA=0
      DO 2 I=1,NN,NCRE
      IB=IA+6
      IC=IB/6
      ID=IB-IC*6
      IE=ID*6
      IA=IA+1
      CALL FLD(NBUFER(IC),IE,6, 0,ISYM(I))
 2    CONTINUE
      CALL APRNTV(INCX,INCY,NN,NBUFER,INCRX,INCRY)
      INCRY=INCRY+INCX
      INCRX=INCRX+IABS(INCY)
      IF (INCRX .GE. 1024) NFLAG1=0
      RETURN
      END


$ORIGIN         BRAVO,SYSUT2,REW
$IBFTC MOD7A
      SUBROUTINE TRUCK(NCCNT,NPASS     )
      DIMENSION NNACC(12,256),MTAB(11),IPRT(256),IPLOT(256)
      DIMENSION NTBL(400)
      COMMON /LAB1/XBAR(42,12),SIGMA(42,12),ROT(42,12,12)
      COMMON /LAB2/X(12),NSYM(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IXXX,IYYY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSFC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      NFLGXX=0
      NFLAGX=0
      REWIND LT11
      REWIND LT1
      NFLAG1=0
      MFIN=0
      IXIY=IXXX*IYYY
      DO 10 I=1,IYYY
      MTAB(I)=I
 10   CONTINUE
      DO 50 I=1,400
      NTBL(I)=I
 50   CONTINUE
      DO 11 I=1,IYYY
      READ(LT11) (NNACC(I,JJ),JJ=1,NSPS)
      MFIN=MFIN+1
```

```
 11   CONTINUE
      NUP=NSPS-IXXX+1
      NCNT=NCCNT+1
      NFLAG=0
 200  CONTINUE
      III=MTAB(1)
      DO 110 JJ=1,NSPS
      IF (JJ .GT. NUP) GO TO 102
      IJ=JJ
      JI=JJ+IXXX-1
      NZERO=0
      IKNT=0
      ISUM=0
      JIJ=MTAB(1)
      NTEMP= NCCNT+1
      DO 101 I=IJ,JI
      DO 100 JIJ=1,IYYY
      IIJ=MTAB(JIJ)
      IF (NNACC(IIJ,I) .LE. NCCNT .AND. NNACC(IIJ,I) .NE. 0) GO TO 102
      IF (NNACC(IIJ,I)) 102,107,106
 106  IF (NNACC(IIJ,I) .GT. NTEMP) NTEMP=NNACC(IIJ,I)
      GO TO 100
 107  NZERO=NZERO+1
 100   CONTINUE
 101   CONTINUE
      IF (NZERO .NE. IXIY) GO TO 105
      DO 103 I=IJ,JI
      DO 104 JIJ=1,IYYY
      NNACC(JIJ,I)=NCNT
 104  CONTINUE
 103  CONTINUE
      NCNT=NCNT+1
      IF (NCNT .GT. 400) GO TO 999
      GO TO 110
 105  CONTINUE
      DO 108 I=IJ,JI
      DO 108 JIJ=1,IYYY
      IF (NNACC(JIJ,I) .EQ. 0) NNACC(JIJ,I)=NTEMP
 108  CONTINUE
      GO TO 110
 102  CONTINUE
 110   CONTINUE
      DO 111 JJ=1,NSPS
      IF (JJ .EQ. NSPS ) GO TO 111
      IF (NNACC(III,JJ) .LE. NCCNT) GO TO 111
      IF (NNACC(III,JJ+1) .LE. NCCNT) GO TO 111
      IF (NNACC(III,JJ) .LE. 0) GO TO 111
      IF (NNACC(III,JJ+1) .LE. 0) GO TO 111
      IF (NNACC(III,JJ) .EQ. NNACC(III,JJ+1)) GO TO 111
      IJ=NNACC(III,JJ)
      JI=NNACC(III,JJ+1)
      IF (JI .GT. 400 .OR. IJ .GT. 400) GO TO 111
      IF (NTBL(JI) .GT. NTBL(IJ)) GO TO 125
      NTBL(IJ)=NTBL(JI)
      GO TO 111
```

```
125   CONTINUE
      NTBL(JI)=NTBL(IJ)
111   CONTINUE
1007  FORMAT (1X,I6)
      WRITE(LT1) (NNACC(III,JJ),JJ=1,NSPS)
      IF (MFIN .GE. NSCANS) GO TO 999
      IYI=MTAB(1)
      READ(LT11) (NNACC(IYI,JJ),JJ=1,NSPS)
      MFIN=MFIN+1
      NTEMP=MTAB(1)
      IYY=IYYY-1
      DO 121 I=1,IYY
      MTAB(I)=MTAB(I+1)
121   CONTINUE
      MTAB(IYYY)=NTEMP
      GO TO 200
999   CONTINUE
      DO 122 I=2,IYYY
      III=MTAB(I)
      DO 112 JJ=1,NSPS
      IF (JJ .EQ. NSPS ) GO TO 112
      IF (NNACC(III,JJ) .LE. NCCNT) GO TO 112
      IF (NNACC(III,JJ+1) .LE. NCCNT) GO TO 112
      IF (NNACC(III,JJ) .LE. 0) GO TO 112
      IF (NNACC(III,JJ+1) .LE. 0) GO TO 112
      IF (NNACC(III,JJ) .EQ. NNACC(III,JJ+1)) GO TO 112
      IJ=NNACC(III,JJ)
      JI=NNACC(III,JJ+1)
      IF (JI .GT. 400 .OR. IJ .GT. 400) GO TO 112
      IF (NTBL(JI) .GT. NTBL(IJ)) GO TO 126
      NTBL(IJ)=NTBL(JI)
      GO TO 112
126   CONTINUE
      NTBL(JI)=NTBL(IJ)
112   CONTINUE
      WRITE(LT1) (NNACC(III,JJ),JJ=1,NSPS)
122   CONTINUE
      END FILE LT1
      REWIND LT1
      REWIND LT11
      REWIND LT12
      WRITE(6,1007) (NTBL(I),I=1,400)
      DO 113 I=1,400
      IF (NTBL(I) .EQ. I) GO TO 113
      JI=I+1
      IF (NTBL(JI) .NE. I) GO TO 114
      NTBL(JI)=NTBL(I)
114   CONTINUE
113   CONTINUE
      II=1
      NTEMP=II
      DO 116 I=2,400
      IF (NTBL(I)-NTBL(I-1)) 117,118,119
119   IF (NTBL(I) .NE. I) GO TO 117
      NTBL(I-1)=NTEMP
```

```
      II=II+1
      NTEMP=II
      GO TO 116
118   NTBL(I-1)=NTEMP
      GO TO 116
117   N=NTBL(I)
      NTBL(I-1)=NTEMP
      NTEMP=NTBL(N)
116   CONTINUE
      NTBL(400)=NTEMP
      WRITE(6,1007) (NTBL(I),I=1,400)
      LWER=1
      LOW=NSTART
705   CONTINUE
      NUPPER=LWER+120-1
      NHI=LOW+120-1
      IF (NUPPER .GT. NSPS ) NUPPER=NSPS
      IDIF=NUPPER-LWER+1
      WRITE(6,1005)
1005  FORMAT(1H1)
      CALL LABEL7(LOW,NHI,1)
      DO 710 II=1,NSCANS
      READ(LT1) (NNACC(1,JJ),JJ=1,NSPS)
      DO 115 JJ=1,NSPS
      IB=NNACC(1,JJ)
      IF (IB .LE. 0) GO TO 115
      NNACC(1,JJ)=NTBL(IB)
115   CONTINUE
      IF (NFLGXX .GT. 0) GO TO 127
      WRITE(LT12) (NNACC(1,JJ),JJ=1,NSPS)
127   CONTINUE
      JI=0
      DO 711 JJ=LWER,NUPPER
      JI=JI+1
      N=NNACC(1,JJ)-(NNACC(1,JJ)-1)/45*45+2
      IPRT(JJ)=NSYM(N)
      IPLOT(JI)=NSYM(N)
711   CONTINUE
      WRITE(6,1003) II,(IPRT(JJ),JJ=LWER,NUPPER)
      CALL PLTBF7(IPLOT,IDIF,NBLK,INCX,INCY,NSTX,NSTY,
     1NCRE,NFLAGX,NFLAG1)
1003  FORMAT(4X,I6,1H*,120A1)
710   CONTINUE
      REWIND LT1
      NFLAGX=0
      NFLGXX=1
      NFLAG1=0
      LWER=NUPPER+1
      LOW=NHI+1
      IF (NUPPER .LT. NSPS ) GO TO 705
570   CONTINUE
      NCCNT=NCNT-1
      END FILE LT12
      REWIND LT12
      REWIND LT1
```

```
      IXXX=IXXX-4
      IYYY=IYYY-4
      LT11=13
      RETURN
      END
```

```
$ORIGIN         BRAVO,SYSUT2,REW
$IBFTC MOD7B
      SUBROUTINE SEQMRG(NCLUST,KOUNT,INITCL    )
      COMMON /LAB1/XBAR(42,12),SIGMA(42,12),ROT(42,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IXXX,IYYY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSFC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      DOUBLE PRECISION A(12,12),EIGEN(12,12)
      DIMENSION MERGE(150),MPOP(150),NEXEC(20),C(42,78),B(12,12)
      DIMENSION COM(24)
      EQUIVALENCE(COM(1),NSPS)
 1000 FORMAT(1X,I6,12F10.3)
 1001 FORMAT (1X,4HXBAR    )
 1002 FORMAT(1X,16HDID NOT CONVERG      )
 1003 FORMAT(1X,7HICLUST=   ,I6,14HMERGE(ICLUST)=   ,I6)
 1004 FORMAT (1X,5HRHO=   ,E15.7,5HERR=   ,E15.7)
 1005 FORMAT (1X,12F10.4)
 1006 FORMAT(1X,12I6)
 1007 FORMAT (1X,23HMERGING WILL TAKE PLACE    )
 1008 FORMAT(1H )
 1009 FORMAT (13H COV. MATRIX    )
 1010 FORMAT (12H NORM EIGEN    )
 1011 FORMAT (18H P.A. COV. MATRIX    )
 1012 FORMAT(1H ,6HASUM= ,E15.7,7HCLUSTER,I4)
 1013 FORMAT(1X,28HXBAR(I,J),J=1,12),I=1,KOUNT      )
 1014 FORMAT (1X,29HSIGMA(I,J),J=1,12),I=1,KOUNT      )
 1015 FORMAT(1X,55HROT(I,ICHAN,JCHAN),JCHAN=1,12),ICHAN=1,12),I=1,KOUNT
     1)       )
 1016 FORMAT (1X,I6,(12F10.3))
 1017 FORMAT(1H1)
      NFLG=0
      CZECH=FLOAT(NCHAN)-2.0
      IF (CZECH .LE. 0.0) CZECH=1.0
      REWIND LT10
      REWIND LT12
      RHO=1.0/(10.0**5)
      IF (NSKIP .EQ. 0) GO TO 6
      DO 7 I=1,NSKIP
      CALL SKRBIN(LT10,1,NOP)
 7    CONTINUE
 6    CONTINUE
```

```
      DO 5 ICLUST=1,NCLUST
      MERGE(ICLUST)=ICLUST
5     CONTINUE
      IM=NCHAN
      WRITE(6,1017)
      IN=IM
      IOPT=1
      DO 10 ICLUST=INITCL,NCLUST
      IF (NFLG .GT. 0) GO TO 11
      IF (KOUNT .GE. 42 ) GO TO 11
      KOUNT=KOUNT+1
      IFLAG=KOUNT
      CALL FETCOR(IFLAG,     C,       MPOP,NFLG,INITCL    )
      WRITE(6,1001)
      WRITE(6,1000) IFLAG,(XBAR(IFLAG,I),I=1,12)
      MI=1
      MJ=12
      MK=12
      DO 500 MM=1,12
      WRITE(6,1000) IFLAG,(C(IFLAG,MR),MR=MI,MJ)
      MI=MJ+1
      MJ=MJ+MK-MM
500   CONTINUE
      CALL AMTRX (IFLAG,XBAR,C,A,NCHAN)
      WRITE(6,1008)
      WRITE(6,1009)
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      CALL DJCOBI (A,IM,IN,IOPT,RHO,ERR,EIGEN)
      WRITE(6,1004) RHO,ERR
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      WRITE(6,1008)
      WRITE(6,1005) ((EIGEN(MI,MJ),MJ=1,12),MI=1,12)
      IF (ERR .EQ. 0.0) GO TO 15
      MERGE(ICLUST)=0
      KOUNT=KOUNT-1
      WRITE(6,1002)
      GO TO 10
15    CONTINUE
      CALL ROTA (IFLAG,ROT,EIGEN,NCHAN,A,SIGMA)
      MERGE(ICLUST)=KOUNT
      WRITE(6,1003) ICLUST,MERGE(ICLUST)
      MPOP(KOUNT)=MPOP(ICLUST)
      IF (KOUNT .EQ. 1) GO TO 10
      MCLUST=KOUNT-1
      DO 20 ICHECK=1,20
      NEXEC(ICHECK)=0
20    CONTINUE
      MCHECK=1
      DO 25 JCLUST=1,NCLUST
      IF(MERGE(JCLUST).LT.MCHECK)GO TO 25
      MCHECK=MCHECK+1
      IF (MERGE(JCLUST) .EQ. KOUNT) GO TO 26
      JFLAG=MERGE(JCLUST)
      DO 30 ICHAN=1,NCHAN
      X(ICHAN)=XBAR(JFLAG ,ICHAN)-XBAR(KOUNT,ICHAN)
```

```
 30   CONTINUE
      IFLAG=KOUNT
      CALL KCHFCK(IFLAG,       ROT,X,SIGMA,ASUM,NCHAN)
      WRITE(6,1008)
      WRITE(6,1012)ASUM,JFLAG
      IF (ASUM .GT. CZECH) GO TO 25
      IFLAG=JFLAG
      CALL KCHECK(IFLAG,       ROT,X,SIGMA,ASUM,NCHAN)
      WRITE(6,1008)
      WRITE(6,1012)ASUM,JFLAG
      IF (ASUM .GT. CZECH) GO TO 25
      NEXEC(1)=NEXEC(1)+1
      NSUB=NEXEC(1)+1
      NEXEC(NSUB)=JFLAG
 25   CONTINUE
 26   IF (NEXEC(1) .EQ. 0) GO TO 10
      DO 501 KK=1,NSUB
      WRITE(6,1006) KK,NEXEC(KK)
 501  CONTINUE
      MSUB=NEXEC(1)+1
      TOTAL=MPOP(KOUNT)
      DO 31 IRUN=2,MSUB
      NSUB=NEXEC(IRUN)
      SUM=MPOP(NSUB)
      TOTAL=TOTAL+SUM
 31   CONTINUE
      INUM=0
      DEN=MPOP(KOUNT)
      DO 35 ICHAN=1,NCHAN
      X(ICHAN)=XBAR(KOUNT,ICHAN)*DEN/TOTAL
      DO 40 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      B(ICHAN,JCHAN)=C(KOUNT,INUM)*DEN/TOTAL
 40   CONTINUE
 35   CONTINUE
      DO 45 IRUN=2,MSUB
      NSUB=NEXEC(IRUN)
      INUM=0
      DEN=MPOP(NSUB)
      DO 50 ICHAN=1,NCHAN
      X(ICHAN)=X(ICHAN)+XBAR(NSUB,ICHAN)*DEN/TOTAL
      DO 55 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      B(ICHAN,JCHAN)=B(ICHAN,JCHAN)+C(NSUB,INUM)*DEN/TOTAL
 55   CONTINUE
 50   CONTINUE
 45   CONTINUE
      DO 60 ICHAN=1,NCHAN
      DO 65 JCHAN=ICHAN,NCHAN
      A(ICHAN,JCHAN)=B(ICHAN,JCHAN)-X(ICHAN)*X(JCHAN)
      A(JCHAN,ICHAN)=A(ICHAN,JCHAN)
 65   CONTINUE
 60   CONTINUE
      WRITE(6,1009)
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
```

```
      CALL DJCOBI(A,IM,IN,IOPT,RHO,ERR,EIGEN)
      WRITE(6,1004) RHO,ERR
      WRITE(6,1005) ((A(MI,MJ),MJ=1,12),MI=1,12)
      WRITE(6,1008)
      WRITE(6,1005) ((EIGEN(MI,MJ),MJ=1,12),MI=1,12)
      IF (ERR .NE. 0.0) GO TO 10
      WRITE(6,1007)
      IFLAG=NEXEC(2)
      MPOP(IFLAG)=TOTAL
      INUM=0
      DO 70 ICHAN=1,NCHAN
      XBAR(IFLAG,ICHAN)=X(ICHAN)
      DO 75 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(IFLAG,INUM)=B(ICHAN,JCHAN)
75    CONTINUE
70    CONTINUE
      CALL ROTA (IFLAG,ROT,EIGEN,NCHAN,A,SIGMA)
      DO 80 JCLUST=1,NCLUST
      DO 85 IRUN=2,MSUB
      NSUB=NEXEC(IRUN)
      IF (MERGE(JCLUST) .NE. NSUB) GO TO 85
      MERGE(JCLUST)=IFLAG
85    CONTINUE
80    CONTINUE
      MERGE(ICLUST)=IFLAG
      IF(NEXEC(1).EQ.1)GO TO 94
      ISW=0
      JCHECK=1
      DO 90 JCLUST=1,NCLUST
      IDUM=MERGE(JCLUST)
   91 IF(MERGE(JCLUST).LT.JCHECK)GO TO 90
      IF(MERGE(JCLUST).GT.JCHECK)GO TO 92
      IF(ISW.EQ.1)GO TO 93
      JCHECK=JCHECK+1
      IF(JCHECK.EQ.KOUNT)GO TO 94
      GO TO 90
   92 MERGE(JCLUST)=MERGE(JCLUST)-1
      ISW=1
      GO TO 91
93    IF (JCHECK .GT. KOUNT) GO TO 94
      ISW=0
      INUM=0
      MPOP(JCHECK)=MPOP(IDUM)
      DO 95 ICHAN=1,NCHAN
      XBAR(JCHECK,ICHAN)=XBAR(IDUM,ICHAN)
      SIGMA(JCHECK,ICHAN)=SIGMA(IDUM,ICHAN)
      DO 100 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(JCHECK,INUM)=C(IDUM,INUM)
      ROT(JCHECK,ICHAN,JCHAN)=ROT(IDUM,ICHAN,JCHAN)
      ROT(JCHECK,JCHAN,ICHAN)=ROT(IDUM,JCHAN,ICHAN)
100   CONTINUE
95    CONTINUE
      DO 96 LCLUST=1,NCLUST
```

```
      IF(MERGE(LCLUST).NE.IDUM)GO TO 96
      MERGE(LCLUST)=JCHECK
96    CONTINUE
      JCHECK=JCHECK+1
90    CONTINUE
   94 KOUNT=KOUNT-NEXEC(1)
10    CONTINUE
11    CONTINUE
      WRITE(LT9) (COM(I),I=1,24)
      WRITE(LT9) ((XBAR(I,J),I=1,KOUNT),J=1,12)
      WRITE(LT9) ((SIGMA(I,J),I=1,KOUNT),J=1,12)
      WRITE(LT9) (((ROT(I,ICHAN,JCHAN),I=1,KOUNT),ICHAN=1,NCHAN),
     1JCHAN=1,NCHAN)
      WRITE(6,1013)
      DO 510 I=1,KOUNT
      WRITE(6,1000) I,(XBAR(I,J),J=1,12)
510   CONTINUE
      WRITE(6,1014)
      DO 511 I=1,KOUNT
      WRITE(6,1000) I,(SIGMA(I,J),J=1,12)
511   CONTINUE
      WRITE(6,1015)
      DO 512 I=1,KOUNT
      WRITE(6,1016) I,((ROT(I,ICHAN,JCHAN),JCHAN=1,12),ICHAN=1,12)
512   CONTINUE
      DO 513 I=1,NCLUST
      IF(MERGE(I).GT.KOUNT)GO TO 514
      WRITE(6,515)I,MERGE(I)
  515 FORMAT(1X,7HCLUSTER,I4,1X,5HCLASS,I4)
  513 CONTINUE
  514 CONTINUE
      DO 660 I=1,KOUNT
      DO 620 ICHAN=1,NCHAN
      DO 610 JCHAN=1,NCHAN
      B(ICHAN,JCHAN)=ROT(I,JCHAN,ICHAN)/SIGMA(I,ICHAN)
610   CONTINUE
620   CONTINUE
      DO 650 ICHAN=1,NCHAN
      DO 640 KCHAN=1,NCHAN
      SUM=0.0
      DO 630 JCHAN=1,NCHAN
      SUM=SUM+ROT(I,ICHAN,JCHAN)*B(JCHAN,KCHAN)
630   CONTINUE
      A(ICHAN,KCHAN)=SUM
640   CONTINUE
650   CONTINUE
      WRITE(6,600) I
600   FORMAT(1X,13HCLASS ELLIPSE,I4)
      WRITE(6,1005) ((A(IA,JA),JA=1,NCHAN),IA=1,NCHAN)
660   CONTINUE
      REWIND LT9
      RETURN
      END
```

```
$IBFTF GETCOR
      SUBROUTINE FETCOR(IFLAG,C,NPOP,NFLG,N    )
      COMMON /LAB1/XBAR(42,12),SIGMA(42,12),ROT(42,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IX,IY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSFC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      DIMENSION NPOP(1),C(42,78),NDAT(255)
      DATA NCNT/0/
      INUM=0
      NFLG1=0
      NFLG3=0
      DO 5 ICHAN=1,NCHAN
      XBAR(IFLAG,ICHAN)=0.0
      DO 10 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(IFLAG,INUM)=0.0
 10   CONTINUE
 5    CONTINUE
      KNT=0
 40   CONTINUE
      IF (NCNT .GE. NSCANS) GO TO 70
      NFLG2=0
      READ(LT12)(NDAT(JJ),JJ=1,NSPS)
      NCNT=NCNT+1
      NFLG1=1
      NFLAG2=1
      DO 20 JJ=1,NSPS
      CALL GET7 (X(1),NSPS,0,NCHAN,NSCANO,LT10,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC    )
      IF (NDAT(JJ) .NE. N) GO TO 30
      KNT=KNT+1
      NFLG2=1
      AI=FLOAT(KNT)
      INUM=0
      DO 25 ICHAN=1,NCHAN
      XBAR(IFLAG,ICHAN)=(1.0-1.0/AI)*XBAR(IFLAG,ICHAN)+X(ICHAN)/AI
      DO 26 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      C(IFLAG,INUM)=(1.0-1.0/AI)*C(IFLAG,INUM)+X(ICHAN)*X(JCHAN)/AI
 26   CONTINUE
 25   CONTINUE
      GO TO 20
 30   CONTINUE
      IF (NFLG3 .EQ. 1) GO TO 20
      IF (NDAT(JJ) .NE. N+1) GOTO 20
      NSAV=NCNT
      NFLG3=1
 20   CONTINUE
C     WRITE(6,1000) NCNT,KNT,NSAV,NBCKUP,N,NFLG2,NFLG3
 1000 FORMAT(1X,7I8)
      IF (NFLG2 .NE. 0) GO TO 40
      IF (NFLG3 .EQ. 0) GO TO 40
      NBCKUP=NCNT-NSAV+1
```

```
      CALL BSRFCD(LT10,NBCKUP*I4,RF)
      CALL BSRFCD(LT12,NBCKUP,RF)
      NCNT=NSAV-1
      NPOP(N)=KNT
      N=N+1
      RETURN
 70   CONTINUE
      REWIND LT10
      REWIND LT12
      NFLG=1
      RETURN
      END


$IBFTF TRXAM
      SUBROUTINE AMTRX(IFLAG,XBAR,C,A,NCHAN)
      DIMENSION XBAR(42,12),C(42,78)
      DOUBLE PRECISION A(12,12)
      INUM=0
      DO 1 ICHAN=1,NCHAN
      DO 2 JCHAN=ICHAN,NCHAN
      INUM=INUM+1
      A(ICHAN,JCHAN)=C(IFLAG,INUM)-XBAR(IFLAG,ICHAN)*XBAR(IFLAG,JCHAN)
      A(JCHAN,ICHAN)=A(ICHAN,JCHAN)
 2    CONTINUE
 1    CONTINUE
      RETURN
      END


$IBFTC ATOR
      SUBROUTINE ROTA(IFLAG,ROT,EIGEN,NCHAN,A,SIGMA)
      DIMENSION ROT(42,12,12)
      DIMENSION SIGMA(42,12)
      DOUBLE PRECISION A(12,12)
      DOUBLE PRECISION EIGEN(12,12)
      DO 1 ICHAN=1,NCHAN
      SIGMA(IFLAG,ICHAN)=A(ICHAN,ICHAN)
      DO 2 JCHAN=1,NCHAN
      ROT(IFLAG,ICHAN,JCHAN)=EIGEN(JCHAN,ICHAN)
 2    CONTINUE
 1    CONTINUE
      RETURN
      END


$IBFTF KCEHCK
      SUBROUTINE KCHECK (IFLAG,       ROT,X,SIGMA,ASUM,NCHAN)
      DIMENSION ROT(42,12,12),SIGMA(42,12),X(1)
      ASUM=0.0
      DO 3 ICHAN=1,NCHAN
      SUM=0.0
      DO 4 JCHAN=1,NCHAN
      SUM=SUM+ROT(IFLAG,ICHAN,JCHAN)*X(JCHAN)
 4    CONTINUE
```

```
      ASUM=ASUM+SUM*SUM/SIGMA(IFLAG,ICHAN)
 3    CONTINUE
      RETURN
      END

$ORIGIN         BRAVO,SYSUT2,REW
$IBFTC MOD7C
      SUBROUTINE CLASS(NCLASS,NTEST,NPASS    )
      COMMON /LAB1/XBAR(42,12),SIGMA(42,12),ROT(42,12,12)
      COMMON /LAB2/X(12),ALPHA(49),NSPS,NSCANS,NCHAN,LT9,LT10,LT11,LT12,
     1LT13,LT1,IX,NDUMMY,
     1NSTART,NSTOP,
     1NBTLG,MODE,ITYPE,MSFC,I4,NCRE,
     1NSKIP,INCX,INCY,NSTX,NSTY
      DIMENSION W(12),MTAB(3)
      DIMENSION NDAT(255,3),PRNT(255)
      DIMENSION COM(24)
      EQUIVALENCE(COM(1),NSPS)
      REWIND LT9
      REWIND LT10
      REWIND LT1
      REWIND LT12
      REWIND LT13
      NFLAG1=0
      LT1=1
      IF (NPASS .NE. NTEST) LT1=LT13
      CZECH=NCHAN
      IF (NSKIP .EQ. 0) GO TO 601
      DO 602 I=1,NSKIP
      CALL SKRBIN(LT10,1,NOP)
 602  CONTINUE
 601  CONTINUE
      READ(LT9) (COM(I),I=1,24)
      READ(LT9) ((XBAR(I,J),I=1,NCLASS),J=1,12)
      READ(LT9) ((SIGMA(I,J),I=1,NCLASS),J=1,12)
      READ(LT9) (((ROT(I,ICHAN,JCHAN),I=1,NCLASS),ICHAN=1,NCHAN),
     1JCHAN=1,NCHAN)
      DO 1 I=1,3
 1    MTAB(I)=I
      DO 10 IEND=1,NSCANS
      READ(LT12) (NDAT(I,1),I=1,NSPS)
      NFLAG2=1
      DO 20 ISUBN=1,NSPS
      CALL GET7(X(1),NSPS,0,NCHAN,NSCANO,LT10,IERR,NFLAG2,
     1NSTART,NBTLG,MODE,NCRE,ITYPE,MSFC    )
      IF (NDAT(ISUBN,1) .GT. 0 ) NDAT(ISUBN,1)=0
      SMALL=1.75*CZECH
      DO 25 ICLASS=1,NCLASS
      DO 30 ICHAN=1,NCHAN
      W(ICHAN)=X(ICHAN)-XBAR(ICLASS ,ICHAN)
 30   CONTINUE
      ASUM=0.0
      DO 35 ICHAN=1,NCHAN
      SUM=0.0
```

```
      DO 40 JCHAN=1,NCHAN
      SUM=SUM+W(JCHAN)*ROT(ICLASS,ICHAN,JCHAN)
40    CONTINUE
41    ASUM=ASUM+SUM*SUM/SIGMA(ICLASS,ICHAN)
35    CONTINUE
      IF (ASUM .GT. SMALL) GO TO 25
      SMALL=ASUM
      NDAT(ISURN,1)=ICLASS
25    CONTINUE
1016  FORMAT(1X,3I6,F10.2)
20    CONTINUE
      WRITE(LT1)(NDAT(I,1),I=1,NSPS)
10    CONTINUE
      END FILE LT1
      REWIND LT1
      REWIND LT12
      REWIND LT10
      IF (NPASS .NE. NTEST) GO TO 804
      DO 610 IZ=1,NSCANS
      IY=MTAB(1)
      READ(LT1) (NDAT(IA,IY),IA=1,NSPS)
      NTEMP=MTAB(1)
      MTAB(1)=MTAB(2)
      MTAB(2)=MTAB(3)
      MTAB(3)=NTEMP
      IF (IZ .LT. 3) GO TO 610
      DO 620 IA=NSTART,NSTOP
      IF (IA .EQ. 1) GO TO 620
      IF (IA+1 .GT. NSTOP) GO TO 620
      IIY=MTAB(2)
      IM=MTAB(1)
      IN=MTAB(2)
      M=NDAT(IA,IM)
      N=NDAT(IA-1,IN)
      IF (M .NE. N) GO TO 650
      IL=MTAB(3)
      L=NDAT(IA,IL)
      IF (M .NE. L) GO TO 650
      NDAT(IA,IIY)=M
      GO TO 620
650   IM=MTAB(1)
      M=NDAT(IA-1,IM)
      IN=MTAB(3)
      N=NDAT(IA-1,IN)
      IF (M .NE. N) GO TO 620
      L=NDAT(IA+1,IM)
      IF (M .NE. L) GO TO 620
      NDAT(IA,IIY)=M
620   CONTINUE
      IF (IZ .LT. 3) GO TO 610
      L=MTAB(1)
      WRITE(LT13) (NDAT(I,L),I=1,NSPS)
610   CONTINUE
      DO 611 I=2,3
      L=MTAB(I)
```

```
      WRITE(LT13) (NDAT(IL,L),IL=1,NSPS)
611   CONTINUE
804   CONTINUE
      REWIND LT1
      REWIND LT9
      REWIND LT13
      REWIND LT10
      LOW=NSTART
      LWER=1
800   CONTINUE
      NHI=LOW+120-1
      NUPPER=LWER+120-1
      IF (NUPPER .GT. NSPS ) NUPPER=NSPS
      IDIF=NUPPER-LWER+1
      WRITE(6,1007)
1007  FORMAT(1H1)
      CALL LABEL7(LOW,NHI,1)
      DO 801 II=1,NSCANS
      READ(LT13) (NDAT(JJ,1),JJ=1,NSPS)
      DO 803 JJ=LWER,NUPPER
      IB=NDAT(JJ,1)
      IRND=IB-(IB-1)/45*45+2
      PRNT(JJ)=ALPHA(IRND)
803   CONTINUE
      CALL PLTBF7(PRNT(LWER),IDIF,NBLK,INCX,INCY,NSTX,NSTY,NCRE,
     1NFLAGX,NFLAG1)
      WRITE(6,1008) II, (PRNT(JJ),JJ=LWER,NUPPER)
1008  FORMAT(4X,I6,1H*,120A1)
801   CONTINUE
      REWIND LT13
      NFLAG1=0
      NFLAGX=0
      LWER=NUPPER+1
      LOW=NHI+1
      IF (NUPPER .LT. NSPS ) GO TO 800
802   CONTINUE
      NSCANS=NSCANS-7
      RETURN
      END
```