N71-35321

NASA CONTRACTOR
REPORT

NASA CR-61360

CASE FILE
COPY

SOFTWARE INTERFACING OF FLIGHT DATA

By M. R. Phillips

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

July 1971

Final Report

Prepared for

NASA-GEORGE C. MARSHALL SPACE FLIGHT CENTER
Marshall Space Flight Center, Alabama 35812

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA CR-61360		

4. TITLE AND SUBTITLE	5. REPORT DATE
SOFTWARE INTERFACING OF FLIGHT DATA	July 1971
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S)	8. PERFORMING ORGANIZATION REPORT #
M. R. Phillips	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Engineering Mechanics Division IIT Research Institute 10 W. 35th Street Chicago, Illinois 60616	10. WORK UNIT NO.
	11. CONTRACT OR GRANT NO. NAS8-26209
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D. C. 20546	13. TYPE OF REPORT & PERIOD COVERED CONTRACTOR REPORT
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

16. ABSTRACT

Handling and processing nonsystems flight data storage tapes of vast amounts of data from various installations will require interfacing and data conditioning capabilities and, in some instances, data compression capabilities. These data compression capabilities should reflect the characteristics of the physical data in smaller quantities, such as mean values, correlation functions, spectral density functions and probability density functions.

Flight data received thus far were taken from a multispectral scanner, and have been interfaced to the Marshall Space Flight Center computers. Numerous data sets have been processed, and several algorithms have been checked out using the flight data. The software developed to interface and condition nonsystems raw data tapes was designed to be used also on the UNIVAC 1108 EX8 by modifying the data translation portion only. This software was developed using very modular techniques which provides immediate inclusion of other nonsystems formatted tapes. The flow diagram, Figure 1, shows the input/output, data conditioning, and in general, the data flow.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Flight Data Interfacing Data Plotting Joint Probability Density Function	Unclassified-Unlimited E. D. Geissler E. D. GEISSLER, Dir, Aero-Astrodynamics Laboratory, MSFC

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	70	$3.00

MSFC - Form 3292 (May 1969)

FOREWORD

This final technical report, entitled "Software Interfacing of Flight Data", presents the results of 12 months programming and data processing efforts undertaken by IIT Research Institute (IITRI) from 9 June 1970 to 9 June 1971. Programming and data processing were performed for National Aeronautics and Space Administration (NASA), George C. Marshall Space Flight Center, Alabama 35812, under Contract NAS8-26209 (IITRI Project J6209).

CONTENTS

Section		Page
	SUMMARY	1
1.	IBM 7094 AND UNIVAC 1108 INPUT/OUTPUT SOFTWARE INTERFACE FOR FLIGHT DATA NONSYSTEMS TAPES	1
	A. Larsys 1 Format	3
	1. ID Record	3
	2. n Data Records	3
	B. Larsys 2 Format	4
	1. ID Record	4
	2. Data Record	6
	3. End-of-Tape Record	7
2.	OUTPUT FORMATS FOR DATA PREVIEWING AND ANALYSIS OF FLIGHT DATA	7
	A. Raw Data Listing and Plotting	7
	B. Probability Density Function	9
	C. Joint Probability Density Function	10
3.	SOFTWARE INPUT PROBLEM PARAMETERS FOR CALCULATING PRELIMINARY STATISTICAL PARAMETERS	14
	A. Program Flight	14
	B. Flight Data Interfacing	16
	C. Data Plotting	17
	D. Joint Probability Density	18
4.	DATA CONDITIONING	19
	A. Smoothing, Formatting and Merging	19
	B. Boundary Enhancement	22
	C. Rectangular Sampled Correlations	25
	D. Linear Detrending	29
5.	OPTIONAL MLTCOR LINKS	34
	APPENDIX - Program Listings	40

LIST OF ILLUSTRATIONS

Figure		Page
1.	Flow Diagram	2
2.	Data Plot	8
3.	Joint Probability Distribution Plot	13
4.	Subroutine SUMER	20
5.	Subroutines XHOR and XVERT	21
6.	Boundary Enhancement	23
7.	Boundary Enhancement	24
8.	Rectangular Sampled Correlations	26
9.	Subroutine FINDV	27
10.	Subroutine FINDVD	28
11.	Subroutine FINDH	28
12.	Linear Detrending	30
13.	Original and Composite Signals	31
14.	Frequency Discrimination	32
15.	Linear Detrending	33
16.	Functional Diagram of MLTCOR	35
17.	Link 2	36
18.	Link 2 (Concluded)	37
19.	Link 4	38
20.	Link 4 (Concluded)	39

SOFTWARE INTERFACING OF FLIGHT DATA

M. R. Phillips
IIT Research Institute

SUMMARY

Handling and processing nonsystems flight data storage tapes of vast amounts of data from various installations will require interfacing and data conditioning capabilities and, in some instances, data compression capabilities. These data compression capabilities should reflect the characteristic of the physical data in smaller quantities, such as mean values, correlation functions, spectral density functions and probability density functions.

Flight data received thus far were taken from a multispectral scanner, and have been interfaced to the Marshall Space Flight Center computers. Numerous data sets have been processed and several algorithms have been checked out using the flight data. The software developed to interface and condition nonsystems raw data tapes was designed to be used also on the UNIVAC 1108 EX8 by modifying the data translation portion only. This software was developed using very modular techniques which provides immediate inclusion of other nonsystems formatted tapes. The flow diagram, Figure 1, shows the input/output data conditioning, and in general, the data flow.

1. IBM 7094 AND UNIVAC 1108 INPUT/OUTPUT SOFTWARE INTERFACE FOR FLIGHT DATA NONSYSTEMS TAPES

Digital data tape conversion and reformatting is required to interface tapes received from the Laboratory for Agricultural Remote Sensing (LARS) at Purdue University. Programs have been written to translate and convert Purdue's LARSYS version 1 and version 2 Data Storage Tape (DST). These programs were kept entirely independent of each other. Version 1 will ultimately become obsolete and this module can be removed from the main program without program modifications.

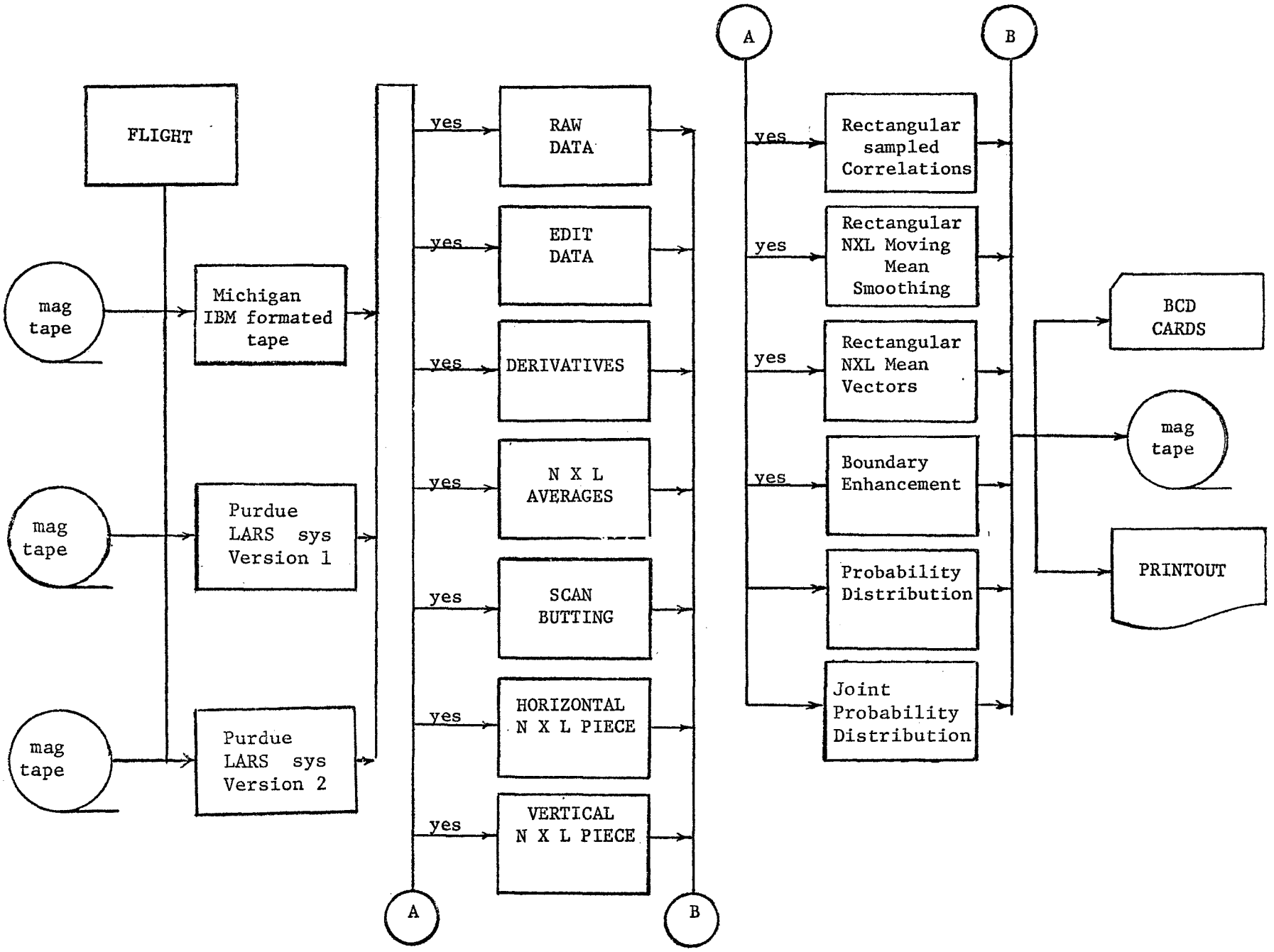


Figure 1 Flow Diagram

A. LARSYS 1 Format

The record blocks on the tape in order are Identification (ID) Record, n data records, and two file marks indicating the end of data. The format of these records is as follows:

1. ID Record.-This record is written as a binary record from a core array ID. The array is 35 words (32 bits per word) with the following information contained in each word:

Word	Information	Date
1	LARS tape number	102
2	Run number	26600061
3	Flight line code	C1
4	Not used	0
5	Number of data channels	12
6	Number of data samples per channel	228
7	Year data were taken	66
8	Month data were taken	06
9	Day data were taken	28
10	Time data were taken (EST)	1229
11	Altitude when data were taken (ft)	2600
12-23	Low cutoff points for spectral channels in order in microns X 100.0	40,44,46, 48,50,52, 55,58,62, 66,72,80
24-35	High cutoff points for spectral channels in order in Microns X 100.0	44,46,48, 50,52,55, 58,62,66, 72,80,100

NOTE: Data from Channel 1 are from the 0.40μ to 44μ wavelength band.
All ID words are integers except ID(3) which is in alphanumeric format (EBCDIC).

2. n Data Records.-Each data record following the ID record contains data from one scan line of data. There are approximately 950 scan lines of data on the tape. There are 2,740 bytes in each data record. The first four bytes (eight bits per byte) is an integer representing the scan line number. The

remaining bytes are data: Byte 5 (B5) is sample 1 from Channel 1 ($S_{1,1}$), B6 is sample 1 from Channel 2 ($S_{2,1}$),..., B16 = $S_{12,1}$, B17 = $S_{1,2}$,..., B2740 = $S_{12,228}$. All data have values in the range 0 to 225 with no sign included in the eight bits. $S_{m,n}$ for all m = 1,12 and n = 1,222 are data from the field of view of the scanner. $S_{m,223}$ is a "0" or dark level data point; $S_{m,224}$ is a variance indicating the variance of the data points used in calculating $S_{m,223}$. The pairs $S_{m,225}$, $S_{m,226}$, and $S_{m,227}$, $S_{m,228}$ are similar to the pair $S_{m,223}$, $S_{m,224}$ except they are from calibration pulses "C1" and "C2" respectively from the scanner.

B. LARSYS 2 Format

There are four types of physical records on the Aircraft Data Storage Tapes:

- ID record - 200 full words fixed length
- Data record - variable length
- End-of-Tape records - 200 full words fixed length
- End-of-File records - IBM standard

An Aircraft Data Storage Tape contains one or more data runs consisting of an ID record, several data records and an End-of-File record. After the last data run on the tape, an End-of-Tape record and two End-of-File records are written on the tape.

For the purposes of this presentation a word is defined to be 32 bits and a byte to be eight bits. Further details regarding the physical records follow.

1. ID Record (200 full words fixed length.-

Word	Format	Description
ID(1)	Integer	LARS tape number (e.g., 1, 17, 102, etc.)
ID(2)	Integer	File number on this tape
ID(3)	Integer	Run number (eight digits aabbbbcc)
		aa = last two digits of the year data were taken

		bbbb = running serial number for the year data were taken
		cc = uniqueness digits for runs which would otherwise have the same run number
ID(4)	Integer	Continuation code
		ID(4) = 0 means the first line of data follows this ID record
		ID(4) = X means that the data following this ID record are a continuation of a flight line started on tape X
ID(5)	Integer	Number of data channels (spectral bands) on tape (30 maximum)
ID(6)	Integer	Number of data samples per channel per line
ID(7-10)	Alphanumeric (4A4)	Flight line identification (16 characters)
ID(11)	Integer	Month data were taken
ID(12)	Integer	Day data were taken
ID(13)	Integer	Year data were taken
ID(14)	Alphanumeric (1A4)	Time data were taken
ID(15)	Integer	Altitude of aircraft
ID(16)	Integer	Ground heading of aircraft
ID(17-19)	Alphanumeric (3A4)	Date data run was generated on this tape (12 characters)
ID(20-50)	Integer	All zero (to be defined later)
ID(51)	Real	Lower limits in micrometers of first spectral band on tape
ID(52)	Real	Upper limits in micrometers of first spectral bank on tape
ID(53)	Real	The suggested value of "C0" calibration pulse.
ID(54)	Real	The suggested value of "C1" calibration pulse
ID(55)	Real	The suggested value of "C2" calibration pulse

ID(56-200)Real	Repeat of ID(51-55) for ID(5) channels in order of appearance in data records
ID(51-200)Real	= 0.0 if data channels do not exist

2. Data Records.-Each data record will contain one scan line of data from ID(5) channels (previous subsection). The first half word (two bytes) of the record will be the record number. The second half word (two bytes) will be the roll parameter which is a number indicating relative position of the roll of the aircraft for this line of data. If the roll parameter is -32,767, the data for the given line do not exist. If the roll parameter has not been calculated it will be set to 32,767. The fifth byte will be the first sample from the first channel. The sixth byte the second sample from the first channel and so on through ID(6) samples and ID(5) channels. A data record will be ID(5)*ID(6)+4 bytes long.

The data from each channel will be from the field of view of the scanner except the last six bytes. The last six are calibration data in the order of appearance.

- C_0 "0" or dark level
- VC_0 Variance of C_0
- C_1 Calibration source C_1
- VC_1 Variance of C_1
- C_2 Calibration source C_2
- VC_2 Variance of C_2

where C_i = Calibration Value i and VC_i = calculated variance of calibration Value i.

During the reformatting process a record may be bad due to tape errors or other errors. When this occurs the data roll parameter, and calibration points will all be set to 0. On good data records all data and calibration values will be in the range of 0 to 255 (bit form) with no sign included in the eight bits.

A data value of 0 or 255 means that the data point was cut off during the digitization process. Data values then range between 0 and 255 with 0 indicating low relative irradiance and 255 indicating high relative irradiance.

3. End-of-Tape Record.-The End-of-Tape Record is very similar to the ID Record with 200 full words in the following format.

Word	Format	Description
ID(1)	Integer	LARS tape number
ID(2)	Integer	File number on this tape
ID(3)	Integer	Run number = 0
ID(4)	Integer	Continuation code ID(4) = 0 means end of data ID(4) = X means data from previous run are continued on tape X
ID(5-50)	Integer	All zero (may be defined later
ID(51-200)	Real	0.0 (may be defined later)

2. OUTPUT FORMATS FOR DATA PREVIEWING AND ANALYSIS OF FLIGHT DATA

A. Raw Data Listing and Plotting

Related to processing flight data and the outputs from the multiscanner spectrometer, the basic main stream program (FLIGHT), inputs the data and prints, writes and/or punches the raw data to print paper, magnetic tapes, and/or cards. A plotting program was written to plot the raw data for previewing before applying data conditioning techniques. This program plots one scan of raw data per frame and has the option to plot only selected channels and scans (Figure 2). Input to this program is a FORTRAN formatted binary tape with a 15 word ID record and n data records containing the multiscanner outputs. Each logical record constitutes one scan of data containing samples for up to 12 channels.

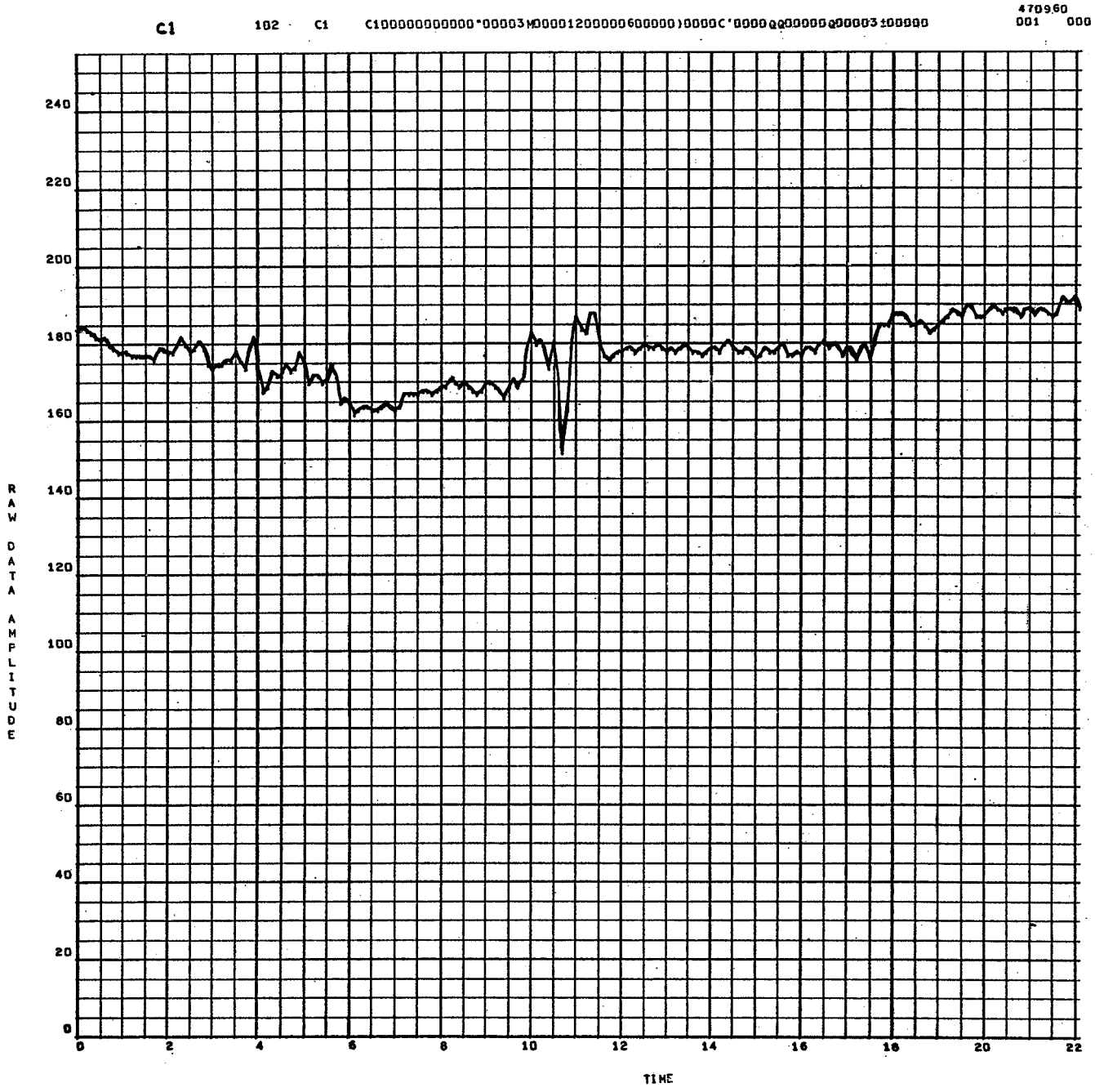


Figure 2 Data Plot

Problem parameters are input with NAMELIST and consist of:

NBEAMS	Number of optical beams, sensors or channels on the input tape
NWHICH	File location of data; maximum of five data files
NFILE	Number of files to be processed; maximum is five
LTN	Logical tape unit of input data
NPLOT	Channel number to be plotted; maximum of 12
NSCANS	Selected scans to be plotted. Maximum of 120 scans if selected; if every scan is desired, set NSCANS = 0 (no maximum)
NSIZE	Physical record size of input data
NO	Number of scans to plot

B. Probability Density Function

A probability density function module was included in the program and is available upon an option selection. This option gives a probability density function of all 12 channels of the multispectral data. This option can be used in conjunction with the rectangular N X L moving mean smoothing module or the N X L rectangular mean vector module.

A probability density function is theoretically related by

$$P(x) = \frac{dP(x)}{dx} \quad , \quad P(x) = \int_{-\infty}^{x} P(\xi)\, d\xi$$

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 computer program finds a set of N data values of x such that the number sequence $\{N_i\}$ satisfies

$$N = \sum_{i=0}^{K+1} N_i$$

where K = number of class intervals.

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

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

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

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

$$\vdots$$

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

As illustrated in Table I this gives a probability density histogram.

This module can be called by any calling program and the necessary problem parameters are passed to the module through calling arguments. These consist of a data list, number of channels, an accumulation table, a look-up table and data index pointers.

C. Joint Probability Density Function

A joint probability density function program has been written to provide a necessary tool in the analysis of the flight data now on hand. This program gives a preliminary insight to the commonality of occurrences observed from two time-series data traces.

The joint probability function P(x,y) is theoretically given by

$$P(x,y) = \frac{\partial}{\partial y}\left(\frac{\partial p(x,y)}{\partial x}\right)$$

where P(x,y) is the probability density function. The probability density function is obtained digitally by selecting an interval $[a,b]$ and the number of subintervals K within that interval.

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

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

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

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.

This program provides a print plot of the number of occurrences at each X and Y coordinate. The densities of the X and Y coordinates are represented by a print character that reflects various shades on the computer print paper. An example of this output is shown in Figure 3.

The joint probability distribution module can be called by any calling subroutine and the problem parameters are passed to the module through its arguments. These consist of a data list, number of channels, number of blocked records per list and an error flag. Additional inputs are included and are input under NAMELIST. These are:

XMAX	Maximum range of data
XMIN	Minimum range of data
IMX	Channel number for the X coordinates
IMY	Channel number for the Y coordinates
RESOL	Number of resolution intervals desired

Figure 3 Joint Probability Distribution Plot

3. SOFTWARE INPUT PROBLEM PARAMETERS FOR CALCULATING PRELIMINARY STATISTICAL PARAMETERS

A. Program Flight

Set Number 1 (NAMDAT):

1.	NBEAMS	Number of beams or sensors Example: NBEAMS = 7; indicates there are seven sensors or seven beams on input tape
2.	*NFLSKC	N/A
3.	*NFLSKD	N/A
4.	*MVMEAN	N/A
5.	*NCEND	N/A
6.	*NOBEAM	N/A
7.	NWHICH	Which files on the input tape to be processed Example: NWHICH - 1,3,4; indicates files 1,3, and 4 only
8.	TIMSAM	Sample rate of input data Example: TIMSAM - .1; indicates data were sampled at 10 samples per sec
9.	*VALMAX	N/A
10.	*VALMIN	N/A
11.	START	Start time on raw data input tape (there can be up to five start times) Example: START = 1.0, 1.5, 2.0, 2.5, 6.0; indicates five starting points on the input tape
12.	STOP	Stop time on raw data input tape (there can be up to five stop times) Example: STOP = 1.4, 1.0, 2.4, 5.9, 6.5; indicates the five stop times for each respective start time in example 11 above
13.	NFILE	Number of files to be processed on the input tape Example: NFILE = 3; indicates three files to be processed
14.	*MSFC	N/A

*These variables are not used in program FLIGHT, but are written on an output used in program MLTCOR.

15.	*IFREQ	N/A
16.	LTN	Logical tape unit input tape is mounted on Example: LTN = 10; indicates tape unit 10 is being used for input
17.	NBLK	Number of records per block on output tape; this variable is used to block the output intermediate tape and to adjust the record size of input tape
18.	*NSTART	N/A
19.	*NBOB	N/A
20.	*PLYBCK	N/A
21.	*NSTEP	N/A
22.	*NDEBUG	N/A

Set Number 2 (COND):

1.	NPIECE	Piece length of data samples desired for input to the correlation program (MLTCOR)
2.	NSCANS	Number of samples or flight scans in the flight line direction desired in data grouping
3.	NSECT	Number of samples perpendicular to the direction of flight desired in data grouping
4.	NDERV	Calculates the derivative of the input data (0 = no; 1 = yes)
5.	NSPAN	Number of samples in a moving mean span if smoothing is desired; = 0, if no smoothing desired
6.	NBUTT	Scan butt option
7.	NCOR	Option to call up the modules for grouping the input data (-1 = vertical direction; 1 = horizontal; 0 = option bypassed)
8.	NSIZE	Physical record size of input tape
9.	IBGN	Beginning sample number in each scan for data grouping
10.	IEND	Ending sample number in each scan for data grouping
11.	NCHAN	Number of channels included in the boundary enhancement option
12.	NSU	This option calls up the boundary enhancement modules
13.	NOWAY	Number of samples separating the operating samples from the reference sample in the boundary enhancement calculations

14. MOVING Moving mean option
15. NPROB Probability density of input data option
16. Editing card with up to 12 values

Problem Setup:

```
$DATA
$COND
NPIECE  =    3
NSCANS  =    3
NSECT   =    3
NDERV   =    1
NSPAN   =    3
NBUTT   =    1
NCOR    =   -1
NSIZE   =  128
IBGN    =    1
IEND    =  167
NCHAN   =    1, 4, 5, 11, 8 * 0
NSU     =    1
NOWAY   =    3
MOVING  =    1
NPROB   =    1
$END
1.274 0.772 0.625 ..... 0.882
7/8
```

B. Flight Data Interfacing

Larsys 1 Format:

LSCANS	Number of scan lines to process
NAME	Identification
NBYTES	Number of eight-bit bytes on nonsystems storage tape
NSKIP	Initial records to skip
NAME2	Identification
NAME3	Identification

Problem Setup:

```
$DATA
$PRUDU1
LSCANS   =   120
NAME     =   6H C1          (ALPHANUMERIC)
NBYTES   =   2664
NSKIP    =   0
NAME2    =   6H FLT 3       (ALPHANUMERIC)
NAME3    =   6H TEST 1      (ALPHANUMERIC)
$END
7/8
```

Larsys 2 Format:

LSCANS	Number of scan lines to process
NSKIP	Initial physical records to skip

Problem Setup:

```
$DATA
$PRUDU2
LSCANS   =   120
NSKIP    =   0
$END
7/8
```

C. Data Plotting

NBEAMS	Number of sensors or channels (Max 12)
NWHICH	Which files wanted plotted
NFILE	Number of files wanted plotted
LTN	Logical tape unit of input tape
NPLOT	Channel number to be plotted
NSCANS	Scan numbers to be plotted
NSIZE	Physical record size of input tape
NO	Total number of scans wanted

Problem Setup:

```
$DATA
$NAMEDAT
NBEAMS   =   12
NWHICH   =   3, 5, 12, 2* 0
NFILE    =   3
LTN      =   9
NPLOT    =   3, 6, 8, 9* 0
NSCANS   =   5, 10, 15, 20, 25, 115* 0
NSIZE    =   256
NO       =   200
$END
7/8
```

D. Joint Probability Density

XMAX	Upper range of data values
XMIN	Lower range of data values
IMX	Location on input tape of X word
IMY	Location on input tape of Y word
RESOL	Number of resolution intervals wanted
ALPNUM	Characters to reflect number of occurrences; up to 11 different characters can be used.

Problem Setup:

```
$DATA
$SCAL
XMAX     =   0.9
XMIN     =   0.4
IMX      =   4
IMY      =   7
RESOL    =   120   (120 is compatible with print paper width.)
.1LFEWM$QRB
$END
7/8
```

4. DATA CONDITIONING

At the present time, data can be input and certain preliminary data conditioning can be performed. Coding is included to edit the raw data to eliminate signal dropouts. This editing is done by inputing replace parameters on cards to correct unwanted values that may appear on the raw data tape.

Coding was included to calculate the first difference of the signal trace for each channel of data to enhance the spectral content of the raw data when correlated. The raw data is then replaced by these differences.

A. Smoothing, Formatting and Merging

Modules were included in the main program to perform data conditioning and handling. Subroutine SUMER was coded to calculate a mean radiance over an N X L matrix of raw data points for each channel of data. The raw data are replaced by these averages and the averages are output to tape and/or cards. An option is provided in the coding to "scan butt" the average data samples to ensure that a data sample obtained from the multispectral scanner, from this particular flight, will always be adjacent to the next sample (Figure 4). In other words, the option makes the data appear as though they were collected from the left of the flight line to the right of the flight line of the first scan, and then from the right of the flight line to the left of the flight line of the second scan, and so on.

Subroutines XHOR and XVERT are subroutines that provide the ability to organize the scan data in such a way as to move through a file of data with an N X L matrix of samples in a horizontal direction (Figure 5a) or in a vertical direction (Figure 5b).

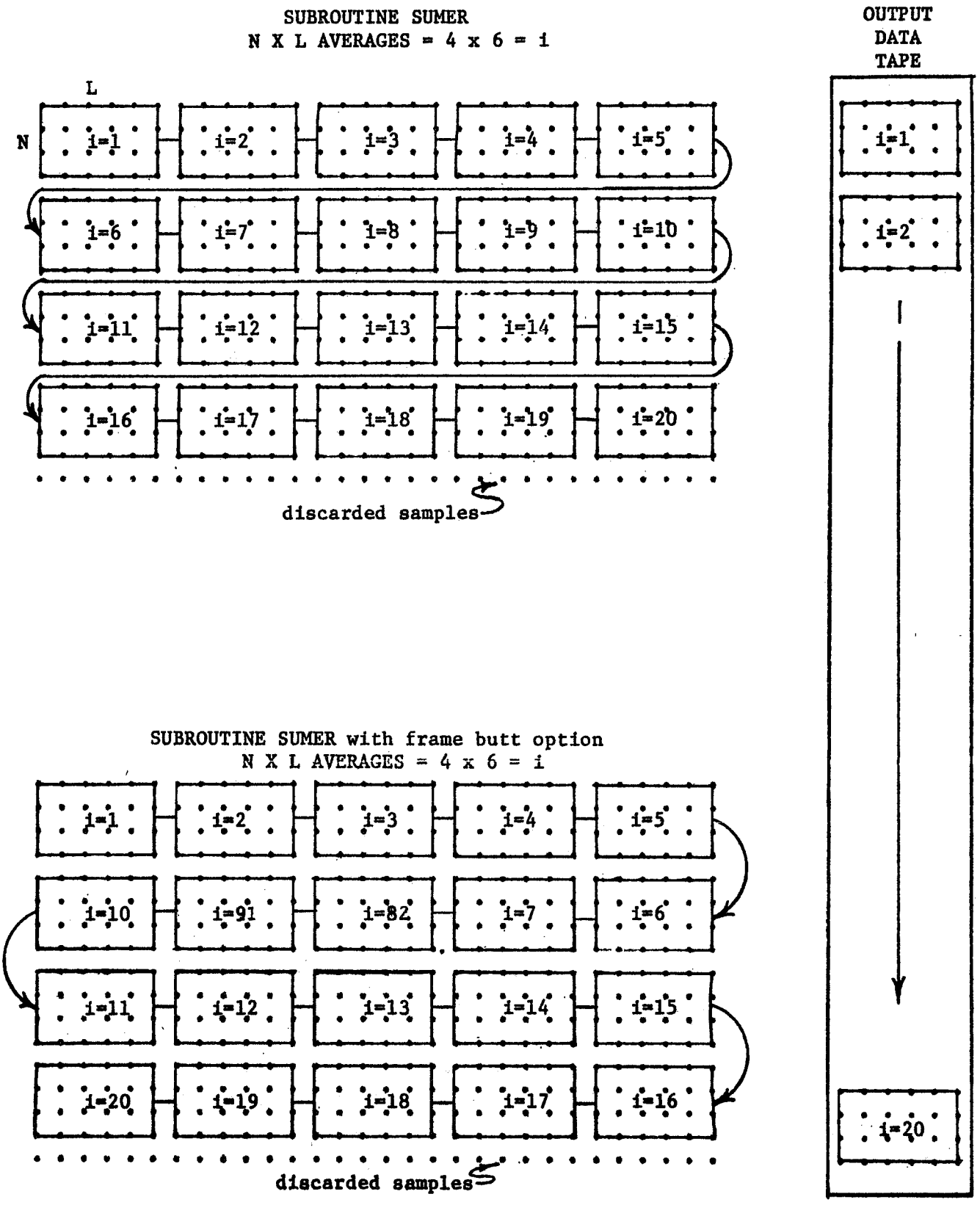


Figure 4 Subroutine SUMER

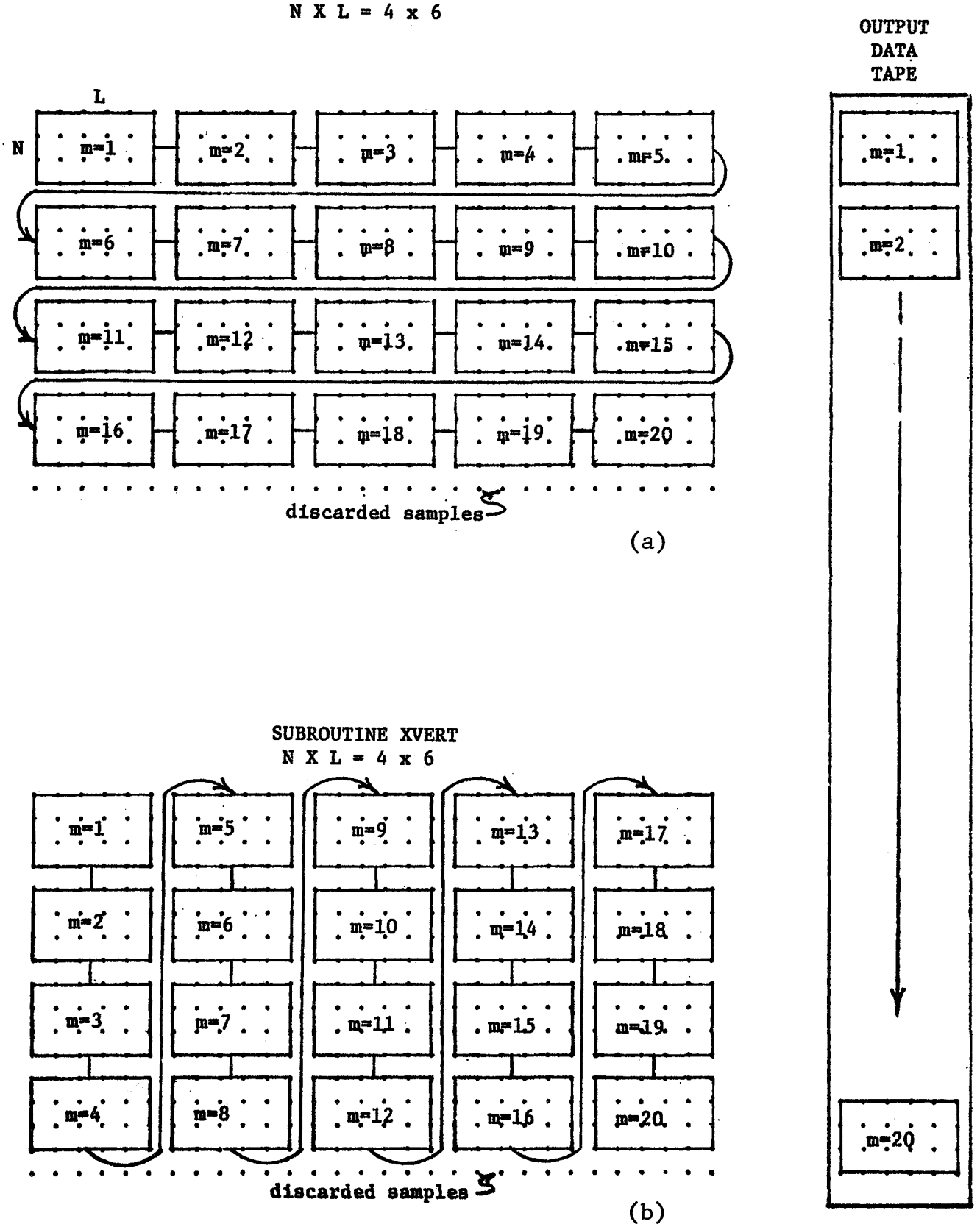


Figure 5 Subroutines XHOR and XVERT

B. Boundary Enhancement

For the purpose of detecting inventory boundaries, a boundary inhancement technique is applied to the raw data or preconditioned data. This enhancement is calculated by taking a two-dimensional array of data and converting the difference between adjacent samples into the square of the difference. The boundary enhancement operator can be extended outward to use, not only the adjacent samples, but also other samples, as illustrated in Figure 3. (Note the letter X indicates the boundary enhancement operator has been moved outward by one sample.)

When the boundary enhancement option is used, the computer program calls up the N X L sample moving mean module to prepare the data. These smoothed data, in a moving N X L rectangular fashion, are input to the boundary enhancement module. All 12 channels are operated simultaneously. This enhances the magnitude of the multispectral data close to the natural inventory boundaries between different homogeneous fields for easier detection of the boundaries (Figure 6).

The 12 channel multispectral data are denoted by $X_{k,ij}$ where k is the channel number, i is the scan line number, and j is the scanwise location of the resolution cell.

The enhanced single channel data are denoted by $y_{ij,\ell}$ where i is the scan line number, j is the scanwise location of the resolution cell, and ℓ is the ℓth sample spaced from the reference sample.

$$
\begin{aligned}
Y_{ij,\ell} = \sum_{k} & \left[X_{k,ij} - X_{k,i(j-\ell)}\right]^2 \\
& + \left[X_{k,ij} - X_{k,i(j+\ell)}\right]^2 \\
& + \left[X_{k,ij} - X_{k,(i-\ell)j}\right]^2 \\
& + \left[X_{k,ij} - X_{k,(i+\ell)j}\right]^2
\end{aligned}
$$

where $\sum_{k}$ denotes the summation of any number of selected channels (Figure 7).

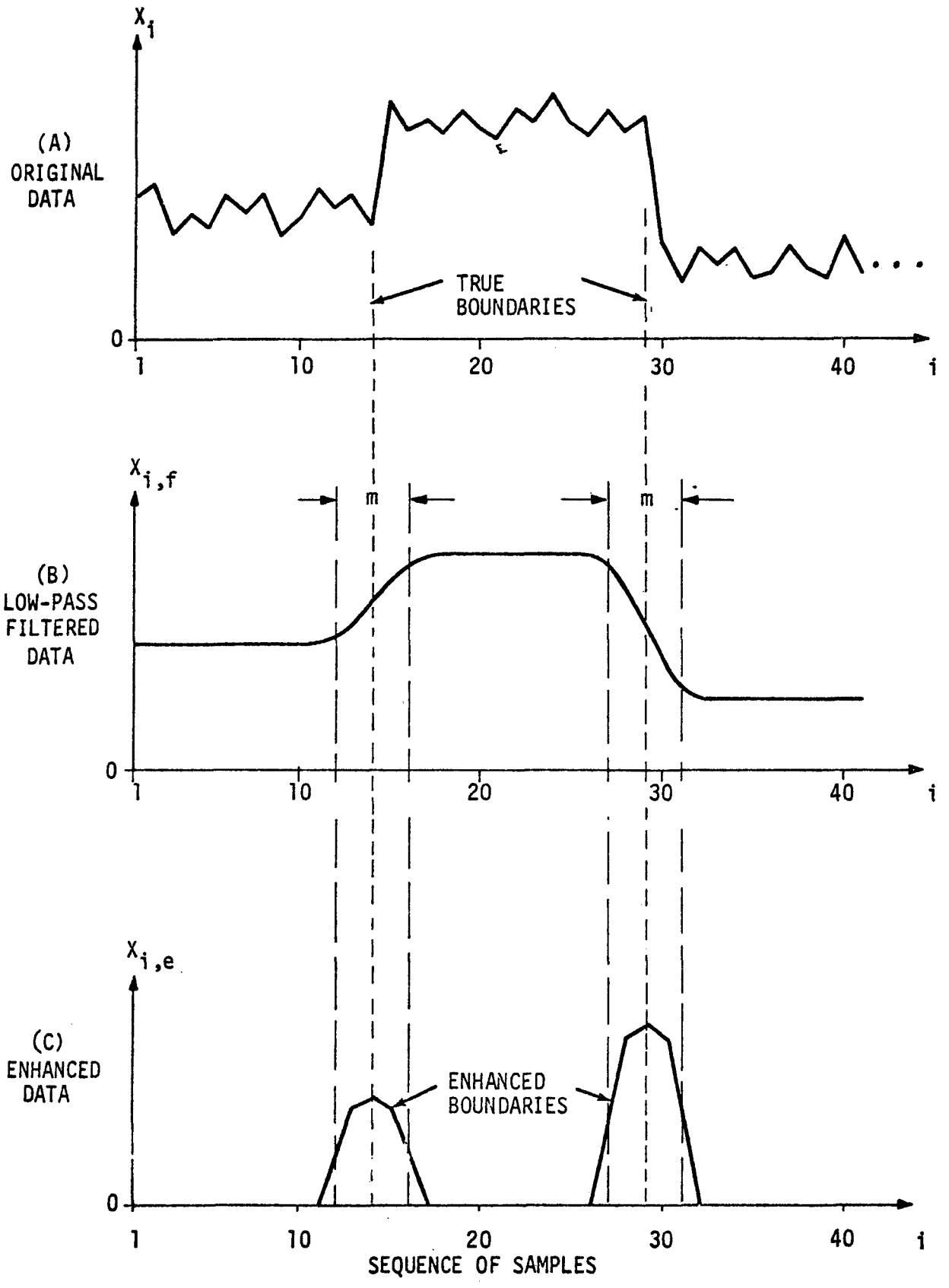


Figure 6 Boundary Enhancement

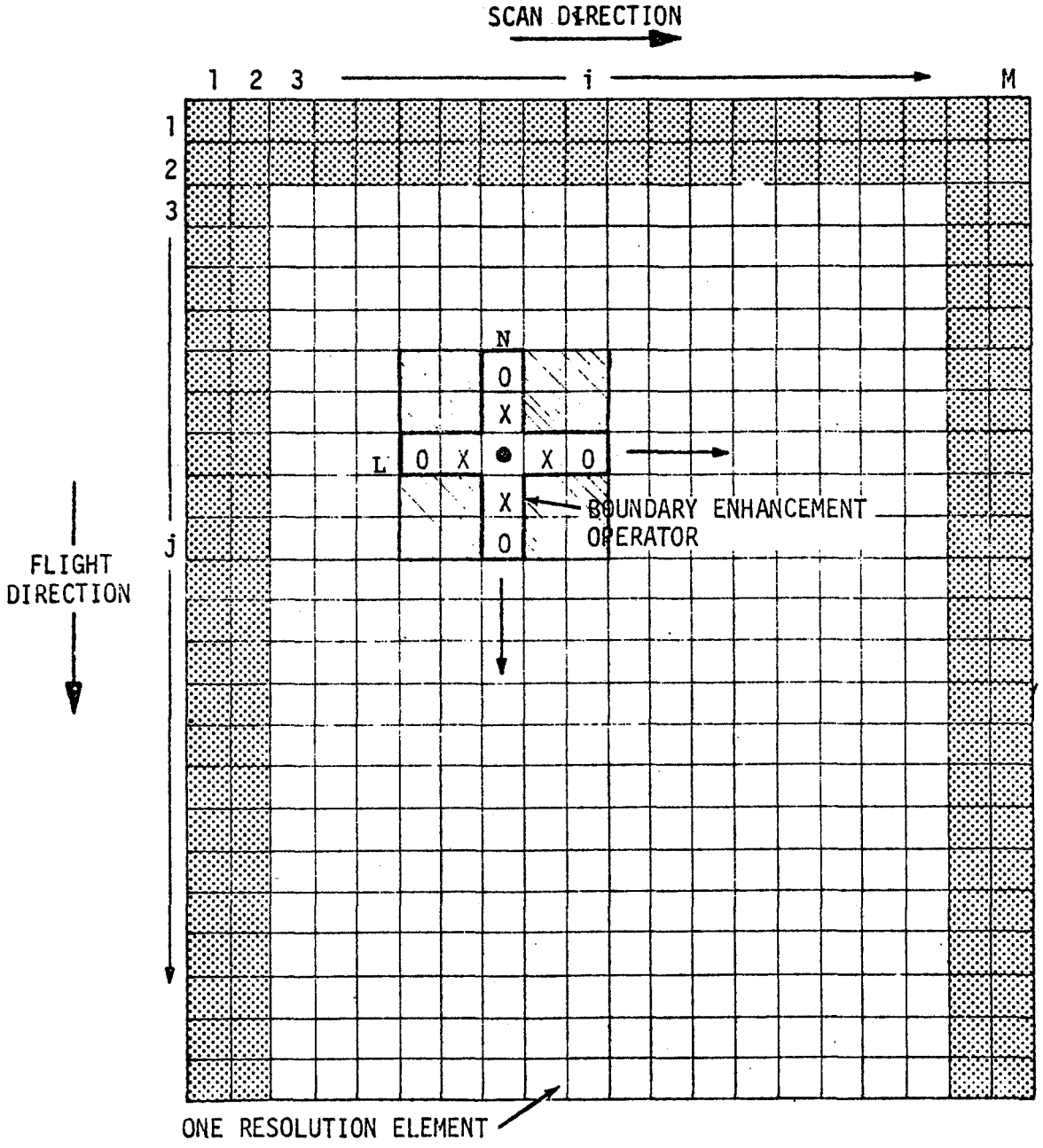
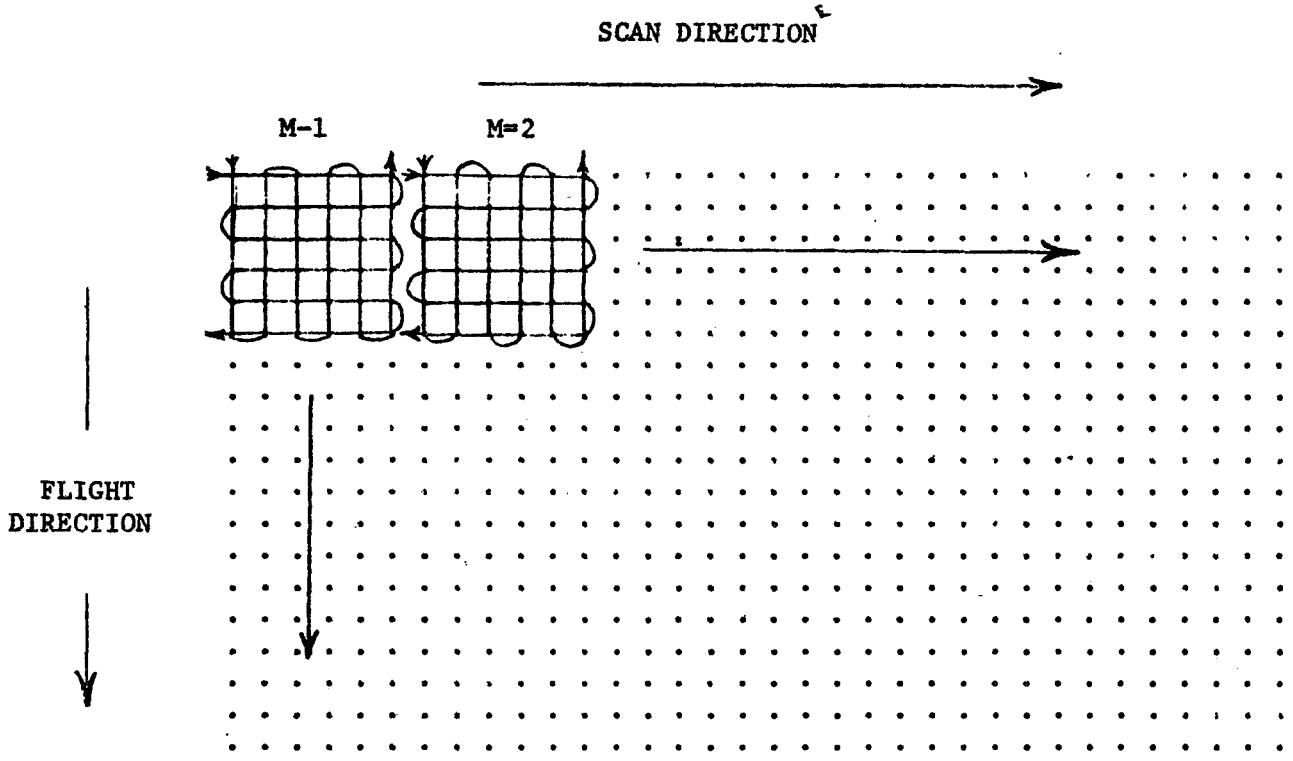


Figure 7 Boundary Enhancement

C. Rectangular Sampled Correlations

Normally correlations are performed on a sequence of time series data, but if rectangular sampled correlations are desired, coding in the computer program generates an output tape that is compatible to the MLTCOR program for calculating correlations. This output tape contains data samples that have been organized into N X L data arrays and each N X L data array constitutes one time series data piece for piecewise correlation calculations. Computer code was added to the MLTCOR computer program to select from the N X L data array the correct data sample to assure a uniform distance between samples for all delayed and undelayed samples and for all time lags. This scheme is illustrated in Figure 8.

Three program modules were written and included in the correlation link of the MLTCOR program, which selects the correct sample to use in the correlations for the vertical, vertical delayed and horizontal samples. When rectangular sampled correlations are wanted, the correlation module passes to the rectangular sampled modules, the location pointer of the sequential time series data point, and through an integer arithmetic calculation based on the N X L rectangle. These location pointers are modified to point to the appropriate location of the sample in the N X L rectangular data array. Program listings of these modules are presented in Figures 9, 10 and 11.



NOTE: The arrows indicate the path and sequence that data samples were selected, for each product of delayed and undelayed samples. The selection of the delayed samples follow the same path as undelayed samples.

Figure 8 Rectangular Sampled Correlations

```
C******THIS MODULE SELECTS THE CORRECT SAMPLE
C******IN THE VERTICAL DIRECTION.
      SUBROUTINE FINDV (NSCANS, NSECT, N, NN)

      IA=N-1
      IB=(IA+NSECT)/NSECT
      IC=IB/2*2
      IF (IB-IC) 51, 50, 51
   51 ID=IA-(IB-1)*NSECT
      NN=IB+ID*NSCANS
      NTEMP=ID
      GO TO 52
   50 NN=IB+NTEMP*NSCANS
      NTEMP=NTEMP-1
   52 CONTINUE
      RETURN
      END
```

Figure 9 Subroutine FINDV

```
C******THIS MODULE FINDS THE CORRECT SAMPLE
C******IN THE DELAYED CHANNEL IN THE VERTICAL
C******DIRECTION.
      SUBROUTINE FINDVD(NSCANS, NSECT, N, NN, IE, NTEMP)
      IA=N-1
      IB=(IA+NSECT)/NSECT
      IC=IB/2*2
      IE=IB-IC
      IF (IE) 51, 50, 51
   51 ID=IA-(IB-1)*NSECT
      NN=IB+ID*NSCANS
      NTEMP=ID
      GO TO 52
   50 NN=IB+NTEMP*NSCANS
      NTEMP=NTEMP-1
   52 CONTINUE
      RETURN
      END
```

Figure 10 Subroutine FINDVD

```
C******THIS SUBROUTINE FINDS THE CORRECT
C******SAMPLE IN THE HORIZONTAL DIRECTION.
      SUBROUTINE FINDH (NSCANS, N, NN)
      ID=(N-1)/NSCANS
      IC=ID/2*2
      IF (ID-IC) 52, 53, 52
   52 IE=(ID+1)*NSCANS
      NN=ID*NSCANS+(IE-N)*1
      GO TO 54
   53 NN=N
   54 CONTINUE
      RETURN
      END
```

Figure 11 Subroutine FINDH

D. Linear Detrending

In the past, data consisting of atmospheric and meteorological parameters have been plagued with unstable and unpredictable boundary changes reflecting unstationary conditions in data sets. These unstable phenomena in some cases completely engulf the usable signal and vital information is lost. In order to retrieve these usable signals, a linear detrending data conditioning technique was developed. This technique is capable of discriminating between a small-scale component and a large-scale component. This is accomplished by calculating a data envelope, bounded by the minimum and maximum peaks of the data trace, and the mean of the data envelope. The fluctuations of the original data about this mean reflect the first generation fluctuations and the detrended small-scale information, (see Figure 12).

To illustrate the small- and large-scale discriminating capabilities of this technique, a theoretical composite signal is broken into four different components by performing the operation four times which decomposes the composite signal to a straight line. This is done by replacing the original signal with the mean of the data envelope and repeating the operation for as many generations desired. (see Figures 13 and 14).

This technique was applied to actual field data consisting of the afore-mentioned problems. A data set was selected from a series of Haswell data runs and two generations of fluctuations were calculated. The first generation retrieved the small-scale information, Figure 15(b), and the second generation retrieved the next larger scale information, Figure 15(d).

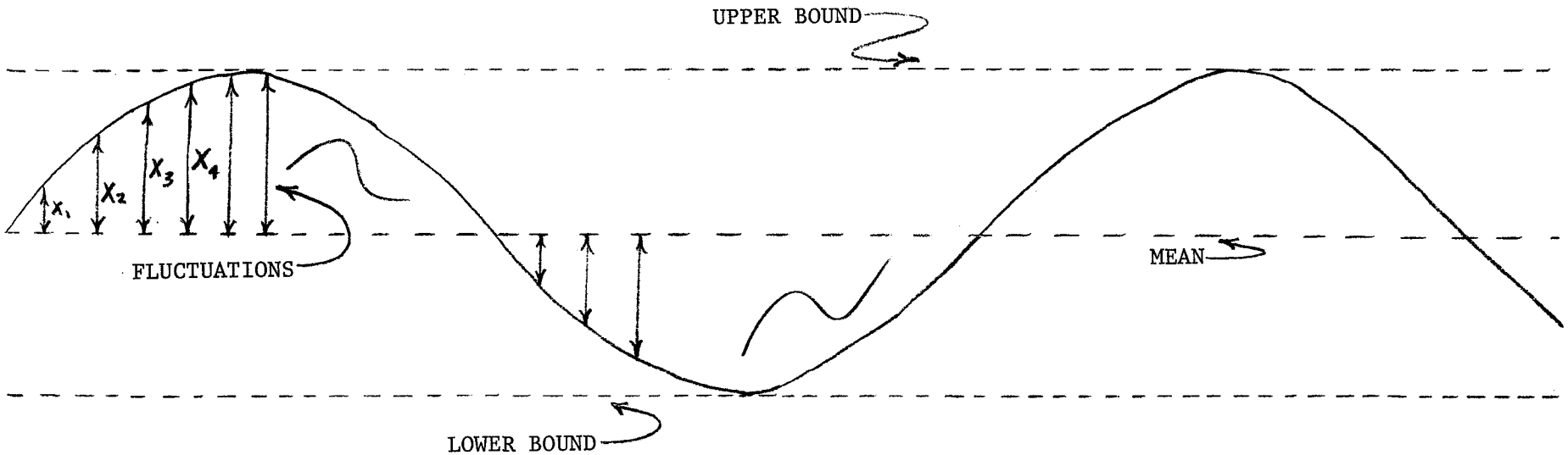
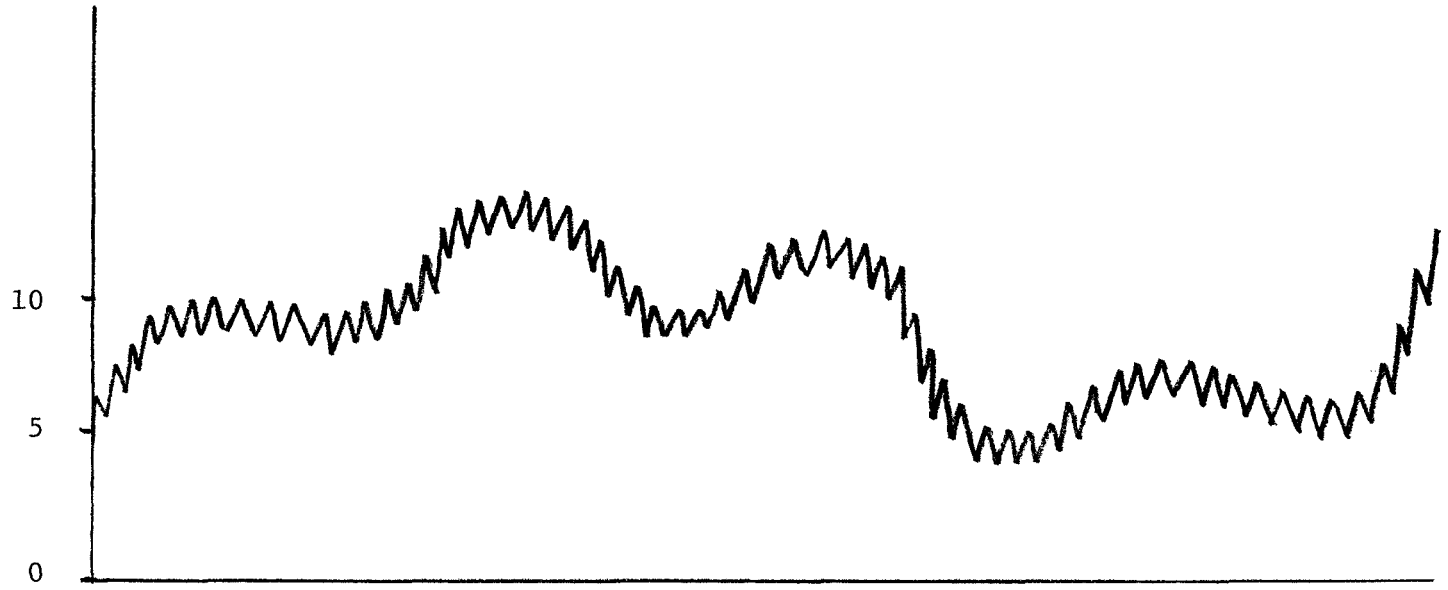
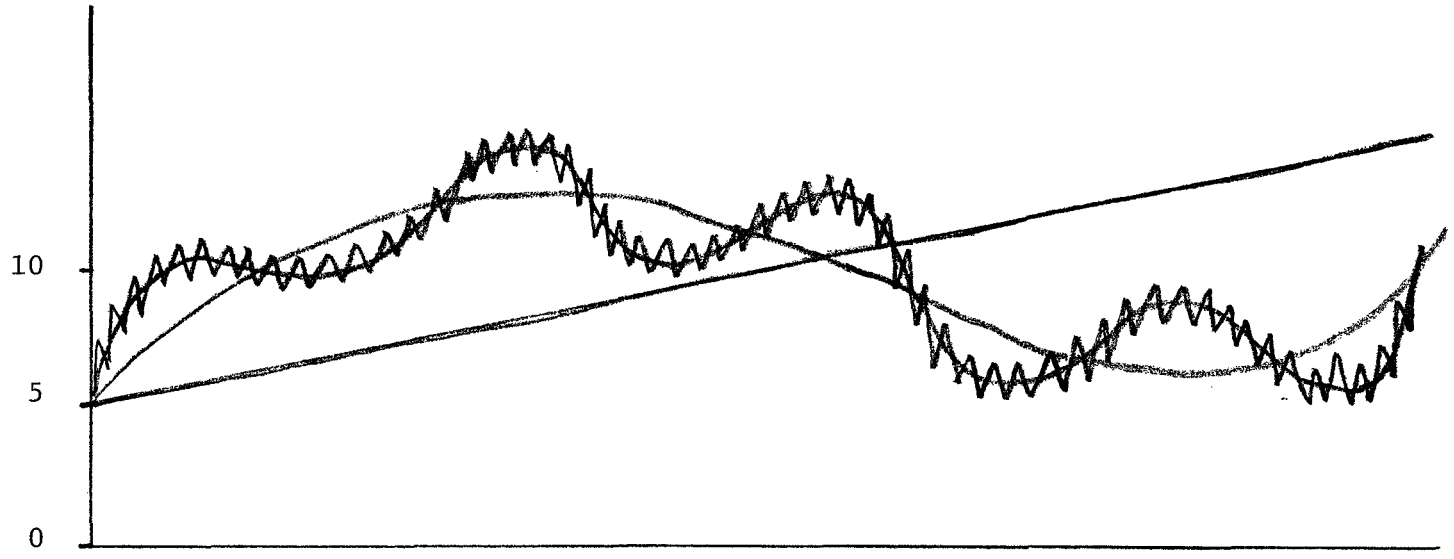


Figure 12 Linear Detrending



ORIGINAL SIGNAL



COMPOSITE SIGNAL

Figure 13 Original and Composite Signals

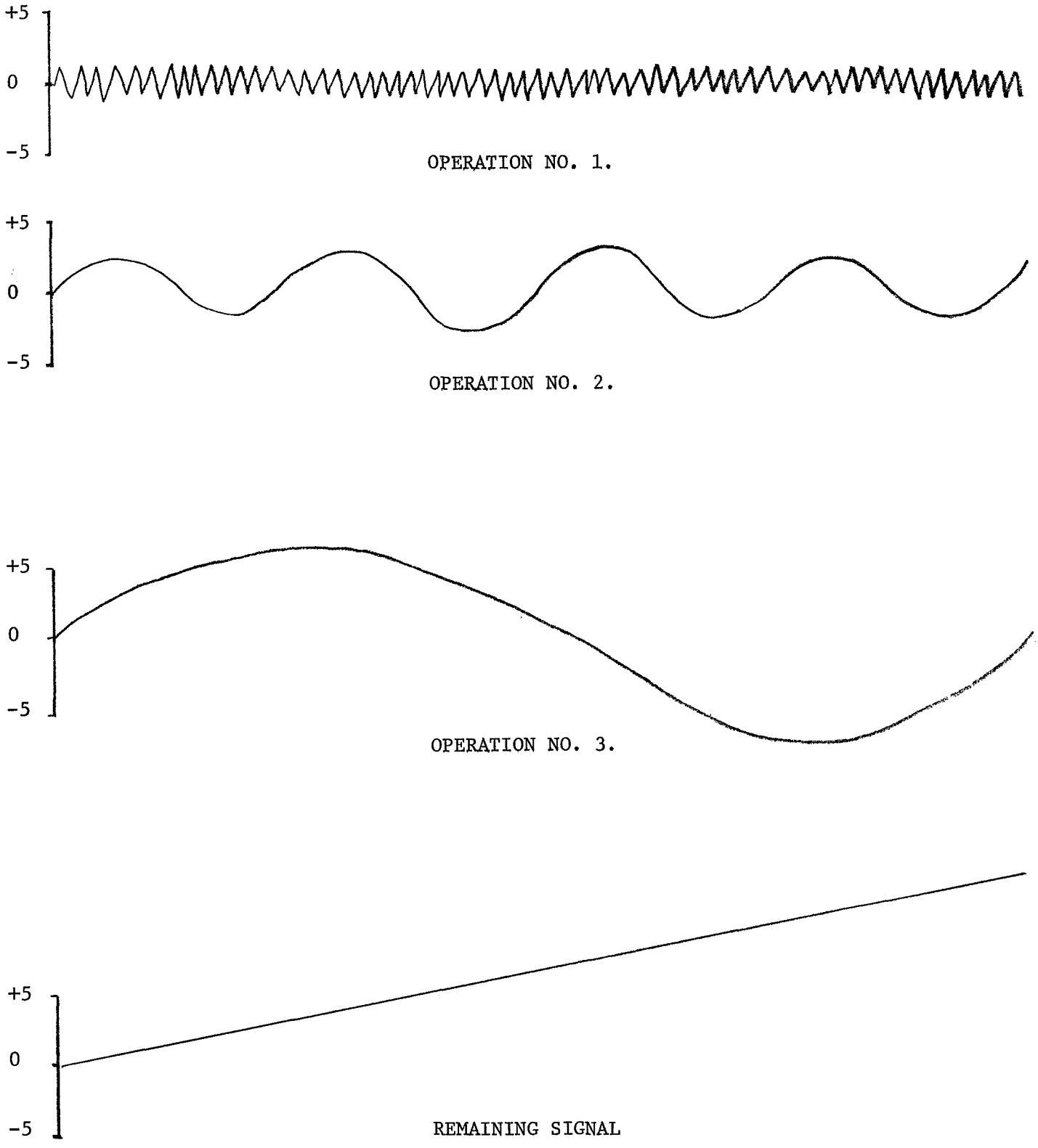


Figure 14 Frequency Discrimination

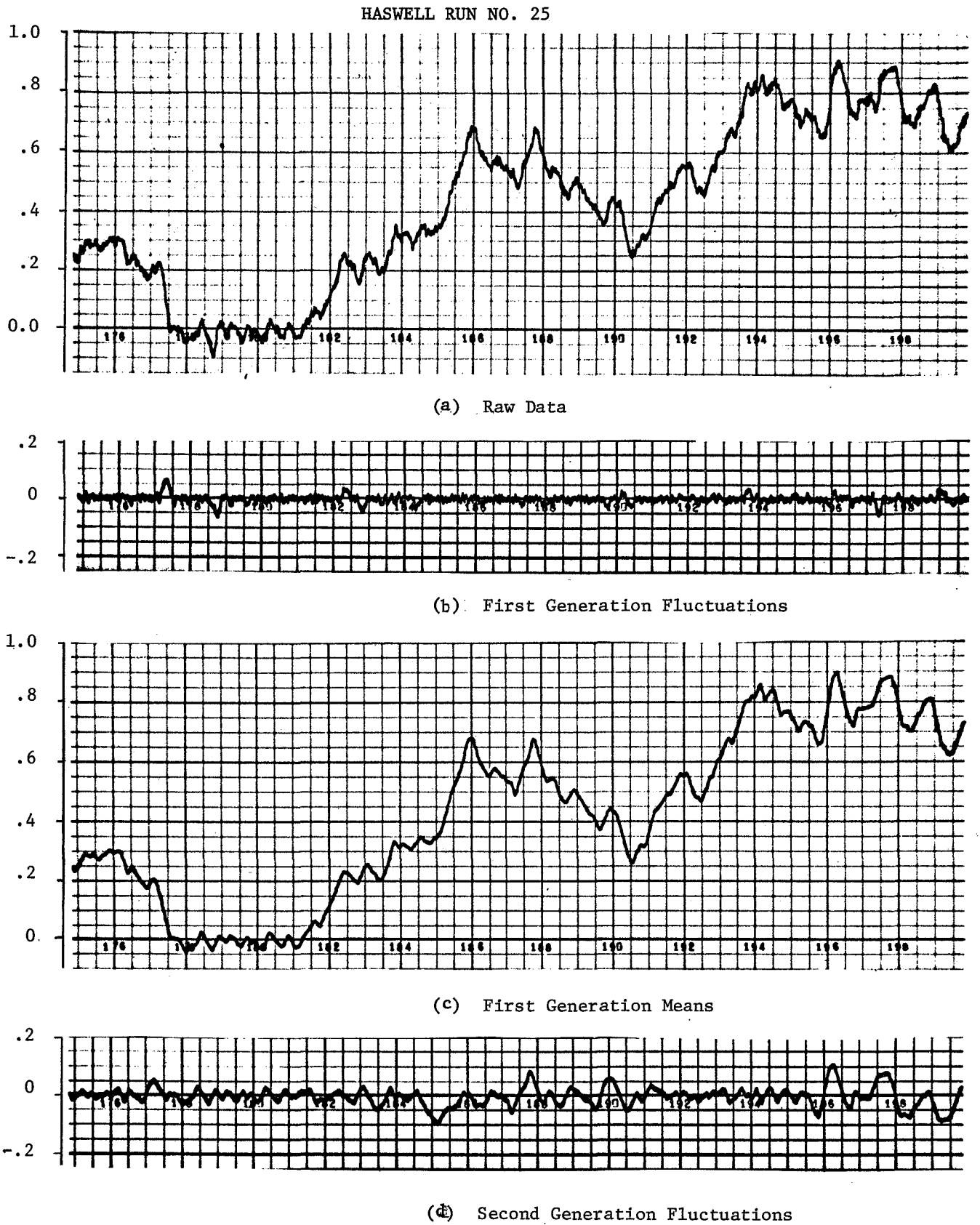


Figure 15 Linear Detrending

5. OPTIONAL MLTCOR LINKS

Several options have been made to MLTCOR to maintain compatibility with the data flow processing and requirements pertaining the NASA Contract NAS8-26209. A general functional diagram of the MLTCOR in its present state is shown in Figure 16. As shown, two additional links (LINK 2 and LINK 4) have been programmed and included in MLTCOR. LINK 2 is the overlay link that applied the linear detrending algorithm to the data. This algorithm decomposes the frequency content of the raw data and the operations can be repeated for as many generations as desired, (Figures 17 and 18).

LINK 4 is the overlay link that classifies the piecewise correlations calculated in LINK 3. Piecewise correlations are read from an intermediate storage tape and each piecewise correlation is classified and written back onto an auxiliary intermediate storage tape with its associated class attached to the data, (Figures 19 and 20).

REFERENCE

1. Bendat, J. S. and Piersol, A. G., Measurement and Analysis of Random Data, John Wiley and Sons, Inc. New York, 1967, Sections 7.2.2 and 7.4.1.

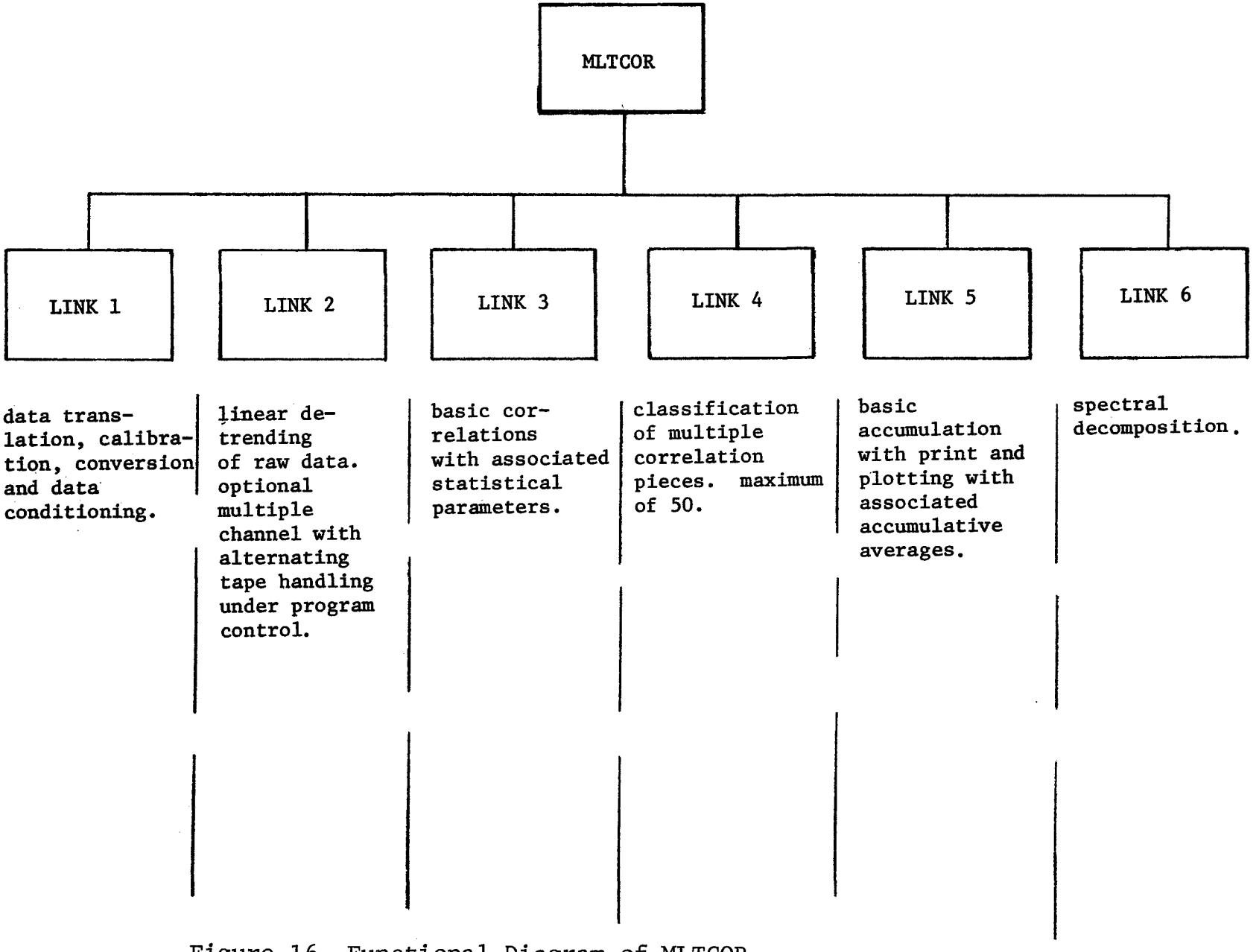


Figure 16 Functional Diagram of MLTCOR

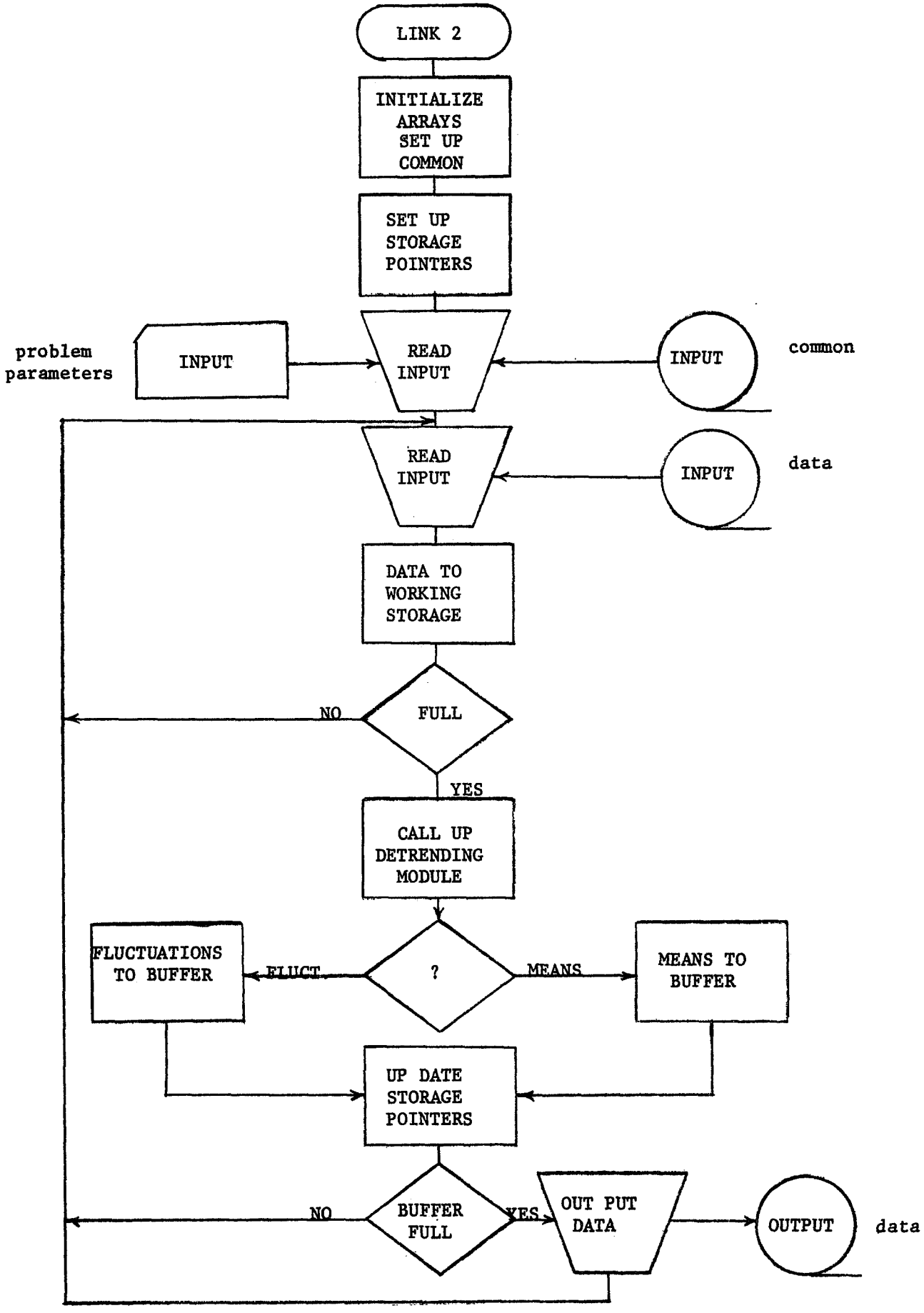


Figure 17 LINK 2

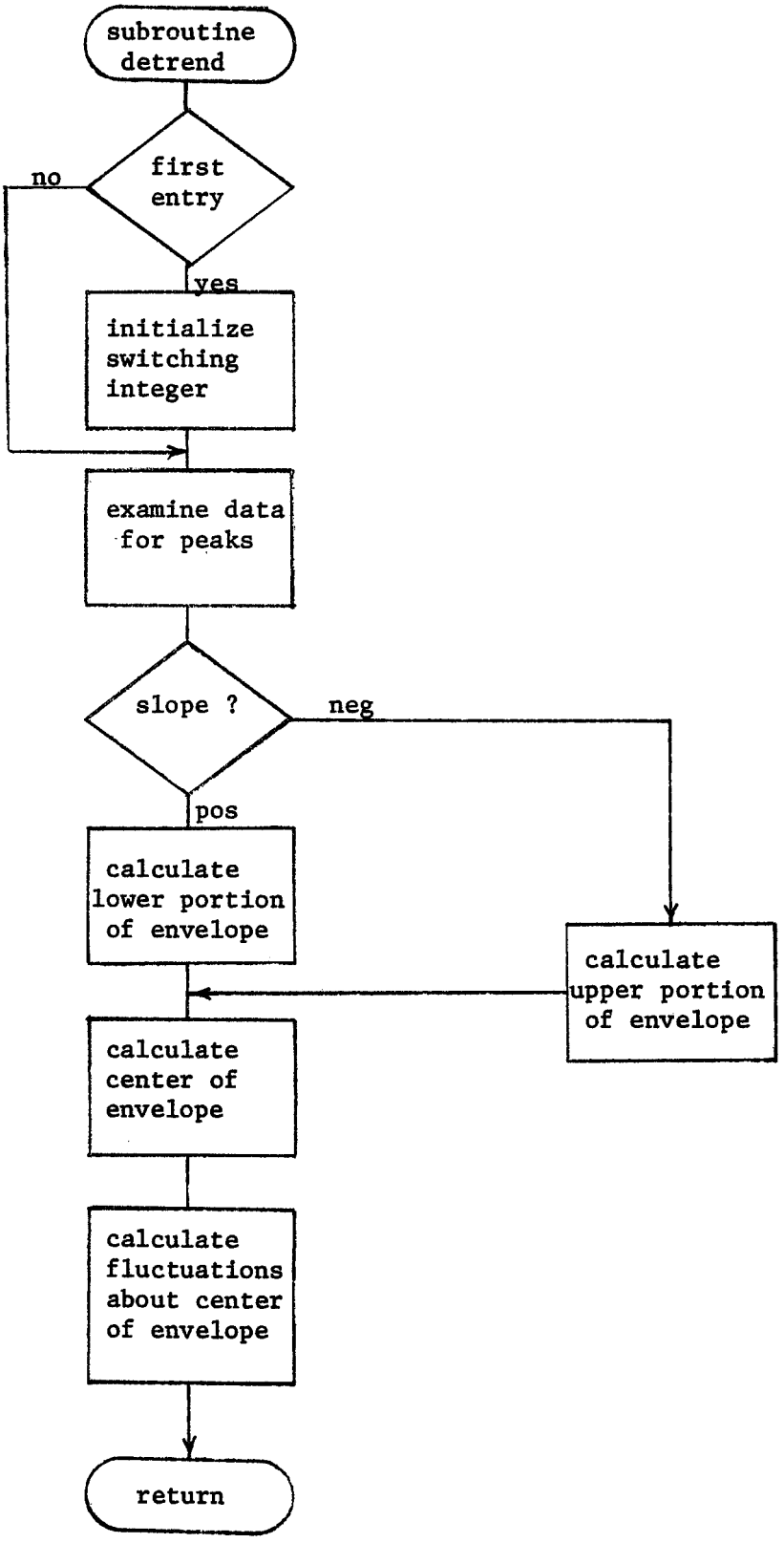


Figure 18 LINK 2 (Concluded)

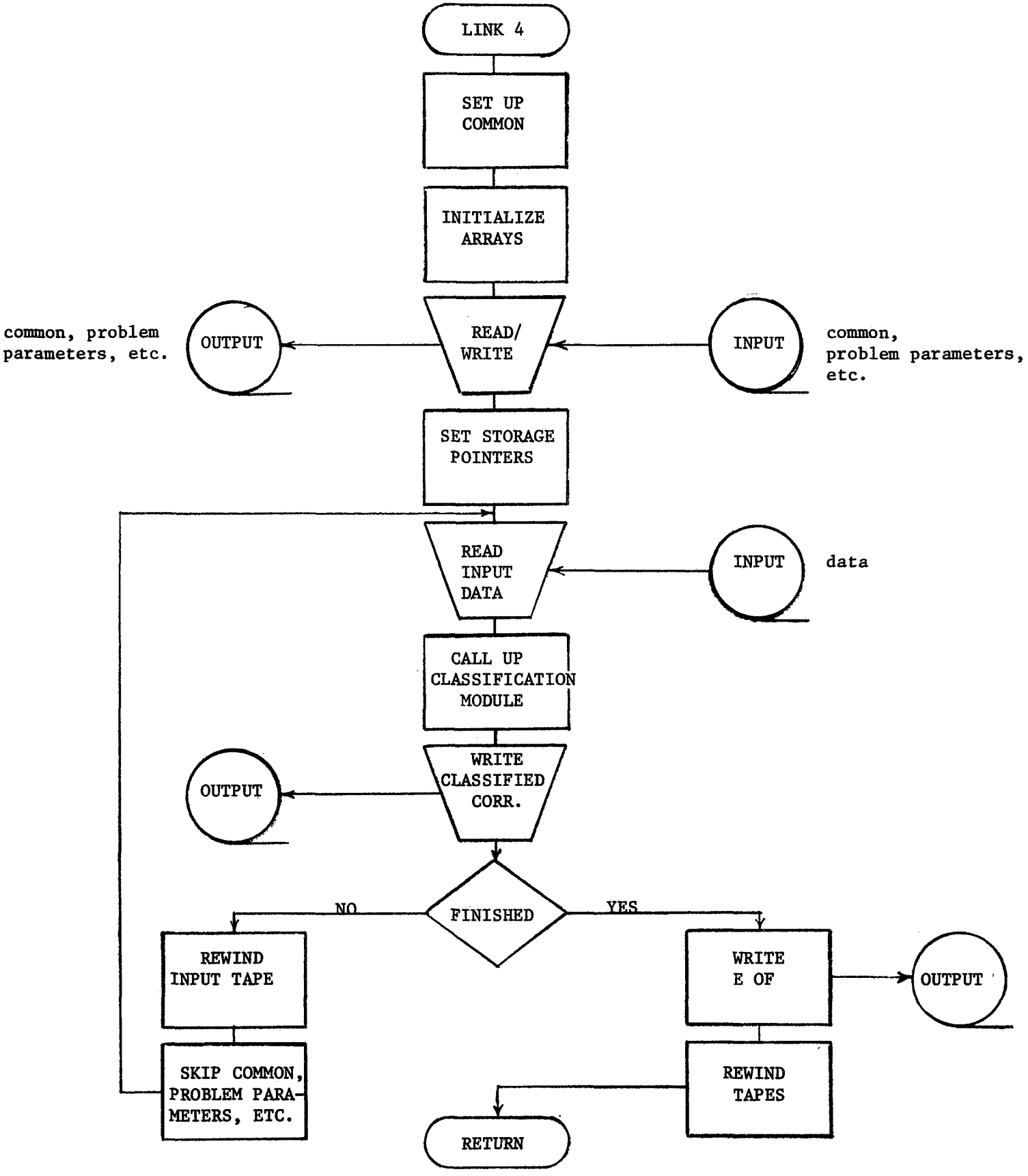


Figure 19 LINK 4

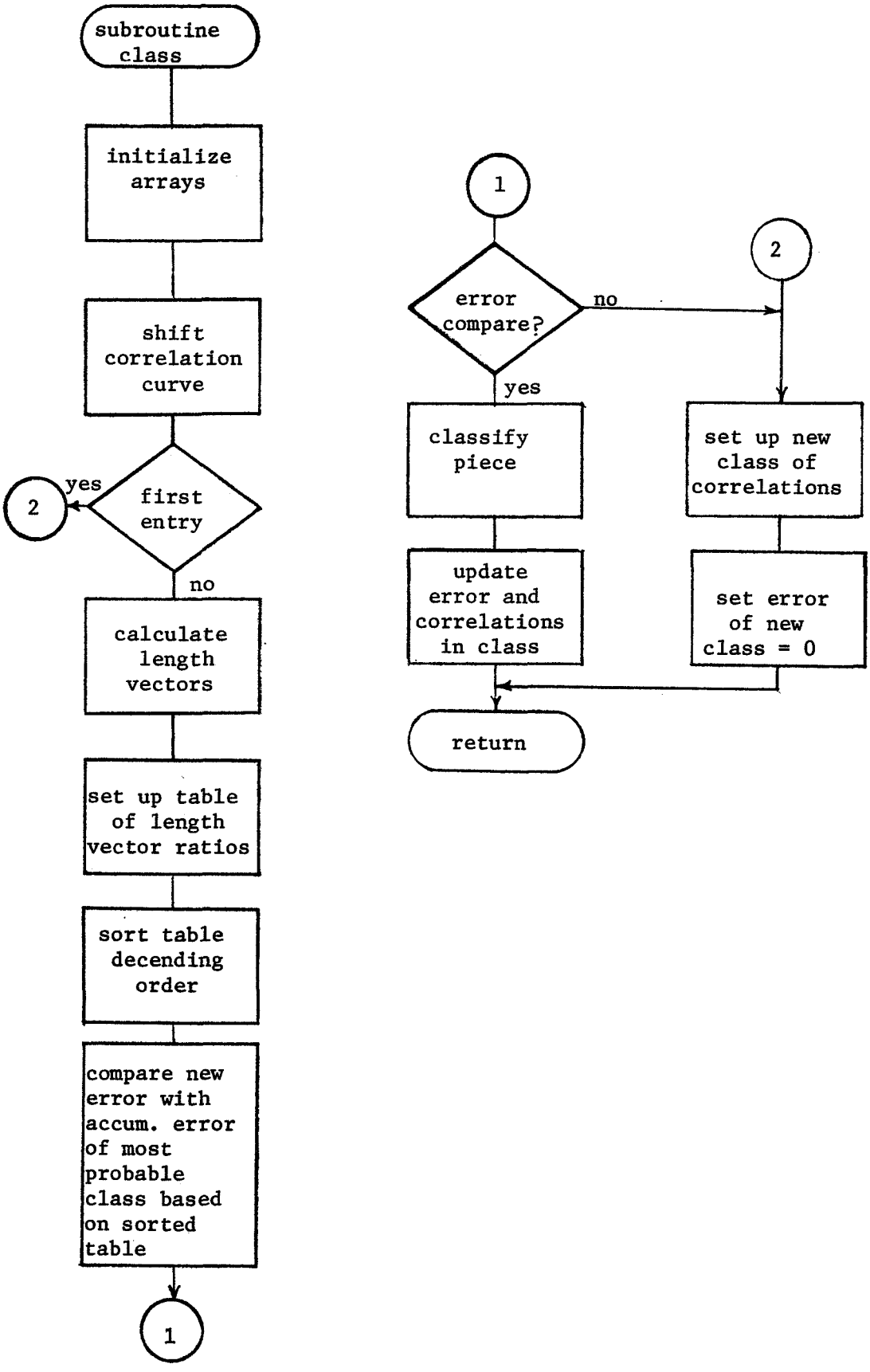


Figure 20 LINK 4 (Concluded)

APPENDIX

PROGRAM LISTINGS

```
      DIMENSION TABLE(101,12),DATAB(101)
      DIMENSION AVGCL(12),R(500),NOFIL(5)
      DIMENSION NWHICH(5),NCHAN(12)
      DIMENSION RCDID(15),COM(175)
      NAMELIST /NAMDAT/ NBEAMS,NFLSKC,NFLSKD,MVMEAN,NCENDF,NOBEAM,
     1NWHICH,TIMSAM,VALMAX,VALMIN,START,STOP,NFILE,MSFC,IFREQ,LTN,NBLK,
     2NSTART,NBOB,PLYBCK,NSTEP,NDEBUG
      NAMELIST /COND/NPIECE,NSCANS,NSECT,NDERV,NSPAN,NBUTT,NCOR,NSIZE
     1,IBGN,IEND,NCHAN,NSU,NOWAY,MOVING,NPROB
      DATA (DATAB(I),I=1,101)/0.000,0.025,0.050,0.075,0.100,0.125,
     10.150,0.175,0.200,0.225,0.250,0.275,0.300,0.325,0.350,0.375,
     20.400,0.425,0.450,0.475,0.500,0.525,0.550,0.575,0.600,0.625,
     30.650,0.675,0.700,0.725,0.750,0.775,0.800,0.825,0.850,0.875,
     40.900,0.925,0.950,0.975,1.000,1.025,1.050,1.075,1.100,1.125,
     51.150,1.175,1.200,1.225,1.250,1.275,1.300,1.325,1.350,1.375,
     61.400,1.425,1.450,1.475,1.500,1.525,1.550,1.575,1.600,1.625,
     71.650,1.675,1.700,1.725,1.750,1.775,1.800,1.825,1.850,1.875,
     81.900,1.925,1.950,1.975,2.000,2.025,2.050,2.075,2.100,2.125,
     92.150,2.175,2.200,2.225,2.250,2.275,2.300,2.325,2.350,2.375,
     12.400,2.425,2.450,2.475,2.500/
      COMMON DATA(2040),DAT(8000)
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12),
     1SLOP(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKD,MVMEAN,TIMSAM,
     2NWDS,IFREQ,MSFC,NFILE,RMS(12),VMEAN(12),NORUN,EXPNO,NOPAGE,
     3AVGMIN(12),AVGMAX(12),NBLK,NSTART,COMTIM
      EQUIVALENCE (RCDID(13),NCOUNT)
      EQUIVALENCE (COM(1),START(1))
      DATA NFLAG/1/
      DO 826 J=1,101
      DO 826 I=1,12
      TABLE(J,I)=0.0
  826 CONTINUE
 1011 FORMAT (1H1,50X,26HPROBABILITY DISTRIBUTION     )
 1012 FORMAT (1HO,130H AMPLITUDE   CH 1       CH 2       CH 3       CH 4
     1   CH 5       CH 6       CH 7       CH 8       CH 9       CH 10      CH
     111      CH 12       )
 1014 FORMAT (1H ,F9.4,12F10.1)
 1004 FORMAT (1H 13F8.4)
C     INPUT NAMELIST DATA
      READ (5,NAMDAT)
      WRITE (6,NAMDAT)
C     INITILIZE CONSTANTS
      READ (5,COND)
      WRITE (6,COND)
      READ (5,1008) (RMS(IJ),IJ=1,NBEAMS)
 1008 FORMAT (12F6.3)
 1009 FORMAT (1X,516)
      NTEST2=-1
      NCTR=0
      IRT=9
      DELT=.1
      NDIV=NPIECE
      IF (NSPAN-1) 814,813,814
  813 NEWPC=NDIV
      GO TO 815
```

```
  814  CONTINUE
       NEWPC=NDIV/NSPAN
  815  CONTINUE
       NW=0
       NDWS=256
       NWDS=NBLK*(NBEAMS+1)
       LIMIT2=1
       TIME=-DELT
       NCT=0
       WRITE (10) (COM(I),I=1,175)
C      POSITION TAPE FILES AND NUMBER OF FILES
       IT=1
       DO 999 IAA=1,NFILE
       REWIND LTN
       IAB=NWHICH(IAA)-1
       IF (IAB) 11,11,12
  12   CONTINUE
       DO 10 IAC=1,IAB
       CALL SKFBIN(LTN,1,RD)
  10   CONTINUE
  11   CONTINUE
C      READ RECORD ONE
       READ (LTN)(RCDID(I),I=1,15)
       CALL FLD (NCT,19,17,1,RCDID(13))
       NCOUNT=NCT
C      WRITE (7,1009) NCOUNT
C      WRITE (7,1008) (RCDID(I),I=1,12)
C1008  FORMAT (12A6)
C1009  FORMAT (5X,I6,40H  DATA ARE SCALED X 1000 RIGHT ADJUSTED )
       NBCK=(NCOUNT*NBEAMS/NSIZE+1)*NSECT
       NSET=(NCOUNT/NDIV)
       NNDIV=NSET*NDIV
       NSKIP=NCOUNT-NNDIV+1
       NSKP=(NSKIP-1)*NBEAMS
       NPROB1=(IBGN-NSKIP)/NSPAN
       IF (NPROB1 .LT. 1) NPROB1=1
       NPROB3=(NPROB1-1)*NBEAMS
       NPROB2=IEND/NSPAN
       IF (NSPAN-1) 816,818,816
  818  NEWCT=NNDIV
       NEWCT1=NCOUNT
       NFAKE=NCOUNT
       GO TO 817
  816  CONTINUE
       IF (MOVING) 824,823,824
  824  CONTINUE
       NEWCT=NNDIV-NSPAN+1
       NEWCT1=NCOUNT-NSPAN+1
       NFAKE=NEWCT
       IRT=11
       REWIND IRT
       GO TO 817
  823  CONTINUE
       NEWCT=NNDIV/NSPAN
       NEWCT1=NCOUNT/NSPAN
       NFAKE=NEWCT
```

```
  817  CONTINUE
       WRITE (6,1000) (RCDID(I),I=1,15)
 1000  FORMAT (1X,12A6,I6,5X,2F8.4)
       NUP=NBEAMS*NCOUNT
       NNC=NCOUNT*NBEAMS+1
       NBCKUP=(NSECT-1)*(NUP/128+1)
       DO 500 NCC=1,NBEAMS
       DAT(NNC)=RMS(NCC)
       NNC=NNC+1
  500  CONTINUE
C      READ BINARY SCAN DATA
       IF (NCOR) 820,1,819
  819  CALL VERT(NSCANS,NSECT,TIME,NSKIP,NTEST2,NSET,DELT,NUP,NDERV,
      1IEND,IBGN)
       GO TO 999
  820  CALL XHOR(NSCANS,NSECT,TIME,NSKIP,NTEST2,NSET,DELT,NUP,NDERV,NBCK,
      1IEND,IBGN)
       GO TO 999
    1  CONTINUE
       DO 811 I=1,NBEAMS
  811  AVGCL(I)=0.0
       ICI=0
       IA=1
   60  CONTINUE
       CALL REDTPC(LTN,1,IERR,NW,NUP,DAT)
       GO TO (50,51,50,50,51),IERR
   51  GO TO 999
   50  CONTINUE
       ICI=ICI+1
C      DO 200 ICC=1,NCOUNT
C      NFRM=ICC*NBEAMS-NBEAMS+1
C      NTO=NFRM+NBEAMS-1
C      J=0
C      DO 201 I=NFRM,NTO
C      J=J+1
C      NDAT(J)=DAT(I)*1000.0
C201   CONTINUE
C      WRITE (7,1010) ICC, (NDAT(II),II=1,12)
C1010  FORMAT (1X,I4,12I5)
C200   CONTINUE
C      GO TO 60
C      DO 81 ICC=1,NCOUNT
       DO 81 ICC=IBGN,IEND
       J=0
       NFRM=ICC*NBEAMS-NBEAMS+1
       NTO=NFRM+NBEAMS-1
       DO 810 I=NFRM,NTO
       J=J+1
       NTEST1=DAT(I)
       IF (NTEST1 .EQ. NTEST2) DAT(I)=RMS(J)
C      GO TO 91
C92    NCTR=NCTR+1
C      GO TO 81
C91    CONTINUE
       AVGCL(J)=AVGCL(J)+DAT(I)
  810  CONTINUE
```

```
   81   CONTINUE
        IF (NSPAN .EQ. 1) GO TO 206
        CALL SUMER (NEWCT,DAT(NSKP+1),NSECT,NSPAN,ICI,NBEAMS,NFLAG,NBUTT,
       1NBCKUP,LTN,IRT,MOVING)
        IF (NFLAG .GT. 1) GO TO 60
  206   CONTINUE
        IF (NPROB .EQ. 0) GO TO 825
        CALL PROB (DAT(NSKP+1),NBEAMS,TABLE,DATAB,NPROB1,NPROB2,NPROB3)
  825   CONTINUE
        NODD=(ICI/2)*2+1-ICI
        DO 80 ICC=NSKIP,NEWCT1
        IF (NBUTT .EQ. 0) GO TO 800
        IF (NODD) 801,800,801
  800   NFRM=ICC*NBEAMS-NBEAMS+1
        GO TO 799
  801   NFRM=(NEWCT1-ICC)*NBEAMS+1
  799   NTO =NFRM+NBEAMS-1
        TIME=TIME+DELT
        DATA(IA)=TIME
        R(IT)=DATA(IA)
        DO 83 I=NFRM,NTO
        IA=IA+1
        IT=IT+1
        IF (NDERV) 91,90,91
   91   CONTINUE
        IV=I+NBEAMS
        DATA(IA)=DAT(IV)-DAT(I)
        GO TO 410
   90   DATA(IA)=DAT(I)
  410   CONTINUE
        R(IT)=DATA(IA)
   83   CONTINUE
        IT=IT+1
        IA=IA+1
        IF (IT .GE. NWDS) GO TO 411
        GO TO 80
  411   CONTINUE
C       WRITE (10) (R(IT),IT=1,NWDS)
C       WRITE (6,1002) (R(IT),IT=1,NWDS)
        IT=1
   80   CONTINUE
        IA=1
        GO TO 60
 1007   FORMAT (1H 12F6.3)
 1002   FORMAT (1H 13F9.4)
 1003   FORMAT (1H 12F9.4)
 1006   FORMAT (1H 2F8.4)
  999   CONTINUE
C          *********************************
        ISUM=(IEND-IBGN+1)*ICI
        XSUM=ISUM
        DO 995 I=1,NBEAMS
        AVGCL(I)=AVGCL(I)/XSUM
  995   CONTINUE
        WRITE(6,1003) (AVGCL(I),I=1,NBEAMS)
C          AVGCL HERE IS MEAN FOR TOTAL SECTION OF FIELD WORKING WITH
```

```
C       **********************************
        END FILE 10
        REWIND 10
        END FILE 11
        REWIND 11
        REWIND LTN
        IF (NSU .EQ. 0) GO TO 997
        CALL SUHERE (NFAKE ,            NOWAY,NUP,NSIZE,NCHAN,NTEST2,
       1IRCDID(2),IRT)
  997   CONTINUE
        IF (NPROB .EQ. 0) GO TO 998
        WRITE (6,1011)
        WRITE (6,1012)
        DO 996 II=1,101
        WRITE (6,1014) DATA8(II),(TABLE(II,JJ),JJ=1,NBEAMS)
  996   CONTINUE
  998   CONTINUE
        STOP
        END
```

```
      SUBROUTINE PURDU1(LTN,IRW)
      DIMENSION ID(35)
      DIMENSION DATA (5000),IDATA(1000),IDI(35)
      NAMELIST /PRUDU1/LSCANS,NAME,NBYTES,    NSKIP,NAME2,NAME3
      READ (5,PRUDU1)
      WRITE (6,PRUDU1)
      XNORM=255.0
      NSCAN0=0
      NSCANS=NBYTES
      NWDS=222
C     TRANSLATE ID
      REWIND IRW
      REWIND LTN
      DO 10 I=1,35
      IDI(I)=0
 10   CONTINUE
       I=0
      II=0
 1000 FORMAT(1H ,18)
      CALL REDTPR (LTN,2,IERR,NW,35,ID    )
 301  CONTINUE
      I=I+1
      II=II+1
      CALL FLD (IDI(II),4,32,0,ID(I))
      IF (II .EQ. 35) GO TO 300
      WRITE (6,1000) IDI(II)
      II=II+1
      CALL FLD(IDI(II),4,4,32,ID(I))
      I=I+1
      CALL FLD (IDI(II),8,28,0,ID(I))
      IF (II .EQ. 35) GO TO 300
      WRITE (6,1000) IDI(II)
      II=II+1
      CALL FLD (IDI(II),4,8,28,ID(I))
      I=I+1
      CALL FLD (IDI(II),12,24,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 35) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,12,24,ID(I))
      I=I+1
      CALL FLD (IDI(II),16,20,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 35) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,16,20,ID(I))
      I=I+1
      CALL FLD (IDI(II),20,16,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 35) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,20,16,ID(I))
      I=I+1
      CALL FLD (IDI(II),24,12,0,ID(I))
      WRITE (6,1000) IDI(II)
```

```
      IF (II .EQ. 35) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,24,12,ID(I))
      I=I+1
      CALL FLD (IDI(II),28,8,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 35) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,28,8,ID(I))
      I=I+1
      CALL FLD (IDI(II),32,4,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 35) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,32,4,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 35) GO TO 300
      GO TO 301
  300 CONTINUE
      IDI(1)=NAME2
      IDI(2)=NAME3
      IDI(3)=NAME
      IDI(13)=0
      CALL FLD (IDI(13),0,18,18,NWDS   )
C     WRITE ID RECORD
      WRITE (10) (IDI(I),I=1,15)
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      CALL SKRBIN (LTN,1,RD)
  401 CONTINUE
  402 CONTINUE
      DO 400 MRP=1,LSCANS
      CALL REDTPR (LTN,2,IERR,NW,1000,IDATA)
      GO TO (11,99,11,11,99),IERR
   99 WRITE (6,1002)
 1002 FORMAT (28H END OF FILE ON INPUT TAPE.     )
      GO TO 999
   11 CONTINUE
      II=0
      K=0
C     TRANSLATE RAW DATA 360/30 8 BIT PER BYTE ORIENTED
      K=K+1
      CALL FLD (NSCAN0,4,32,0,IDATA(K))
  302 CONTINUE
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,4,32,IDATA(K))
      K=K+1
      CALL FLD (IDAT,32,4,0,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      IDAT=0
      II=II+1
      CALL FLD (IDAT,28,8,4,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
```

```
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,12,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,20,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,28,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      K=K+1
      CALL FLD (IDAT,28,8,0,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,8,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,16,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,24,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      GO TO 302
  305 CONTINUE
      WRITE (10) (DATA(I),I=1,2664)
      WRITE (6,1003) NSCANO
C     WRITE (6,1001) (DATA(I),I=1,2664)
      WRITE (6,1001) (DATA(I),I=2665,2736)
 1001 FORMAT (1H ,12F10.6)
 1003 FORMAT (1H ,I6)
  400 CONTINUE
  999 CONTINUE
      REWIND LTN
      END FILE IRW
      REWIND IRW
      RETURN
      END
```

```
      SUBROUTINE PURDU2(LTN,IRW)
      DIMENSION ID(200)
      DIMENSION DATA (7000),IDATA(2000),IDI(200)
      NAMELIST/PRUDU2/LSCANS,NSKIP
      READ (5,PRUDU2)
      WRITE (6,PRUDU2)
      XNORM=1.0
      NSCANO=0
C     TRANSLATE ID
      REWIND IRW
      REWIND LTN
      DO 10 I=1,200
      IDI(I)=0
 10   CONTINUE
       I=0
      II=0
 1000 FORMAT(1H ,I8)
      CALL REDTPR (LTN,2,IERR,NW,200,ID    )
 301  CONTINUE
      I=I+1
      II=II+1
      CALL FLD (IDI(II),4,32,0,ID(I))
      IF (II .EQ. 200) GO TO 300
      WRITE (6,1000) IDI(II)
      II=II+1
      CALL FLD(IDI(II),4,4,32,ID(I))
      I=I+1
      CALL FLD (IDI(II),8,28,0,ID(I))
      IF (II .EQ. 200) GO TO 300
      WRITE (6,1000) IDI(II)
      II=II+1
      CALL FLD (IDI(II),4,8,28,ID(I))
      I=I+1
      CALL FLD (IDI(II),12,24,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,12,24,ID(I))
      I=I+1
      CALL FLD (IDI(II),16,20,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,16,20,ID(I))
      I=I+1
      CALL FLD (IDI(II),20,16,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,20,16,ID(I))
      I=I+1
      CALL FLD (IDI(II),24,12,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      II=II+1
```

```
DECK1B

      CALL FLD (IDI(II),4,24,12,ID(I))
      I=I+1
      CALL FLD (IDI(II),28,8,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,28,8,ID(I))
      I=I+1
      CALL FLD (IDI(II),32,4,0,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      II=II+1
      CALL FLD (IDI(II),4,32,4,ID(I))
      WRITE (6,1000) IDI(II)
      IF (II .EQ. 200) GO TO 300
      GO TO 301
  300 CONTINUE
      NSCANS=IDI(5)*(IDI(6)+6)
      NWDS=((NSCANS+4)*8)/36+1
      IDI(13)=0
      CALL FLD (IDI(13),0,18,18,IDI(6))
C     WRITE ID RECORD
C     WRITE (IRW) (IDI(I),I=1,15)
      WRITE (IRW) (IDI(I),I=1,15)
      IF (NSKIP .EQ. 0) GO TO 402
      DO 401 I=1,NSKIP
      CALL SKRBIN (LTN,1,RD)
  401 CONTINUE
  402 CONTINUE
      DO 400 MRP=1,LSCANS
      CALL REDTPR (LTN,2,IERR,NW,NWDS,IDATA)
      GO TO (11,99,11,11,99),IERR
   99 WRITE (6,1002)
 1002 FORMAT (28H END OF FILE ON INPUT TAPE.     )
      GO TO 999
   11 CONTINUE
      II=0
      K=0
C     TRANSLATE RAW DATA 360/30 8 BIT PER BYTE ORIENTED
      K=K+1
      CALL FLD (NSCAN0,20,16,0,IDATA(K))
  302 CONTINUE
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,4,32,IDATA(K))
      K=K+1
      CALL FLD (IDAT,32,4,0,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      IDAT=0
      II=II+1
      CALL FLD (IDAT,28,8,4,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
```

```
DECK1B

      CALL FLD (IDAT,28,8,12,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,20,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,28,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      K=K+1
      CALL FLD (IDAT,28,8,0,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,8,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,16,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      II=II+1
      IDAT=0
      CALL FLD (IDAT,28,8,24,IDATA(K))
      DATA(II)=FLOAT(IDAT)/XNORM
      IF (II .EQ. NSCANS) GO TO 305
      GO TO 302
  305 CONTINUE
C     WRITE (IRW) (DATA(I),I=1,NSCANS)
      WRITE (IRW) (DATA(I),I=1,NSCANS)
      WRITE (6,1003) NSCAN0
C     WRITE (6,1001) (DATA(I),I=1,NSCANS)
 1001 FORMAT (1H ,12F10.6)
 1003 FORMAT (1H ,16)
  400 CONTINUE
  999 CONTINUE
      REWIND LTN
      END FILE IRW
      REWIND IRW
      RETURN
      END
```

```
      SUBROUTINE VERT(NSCANS,NSECT,TIME,NSKIP,NTEST2,NSET,DELT,
     1NUP,NDERV,IEND,IBGN)
      COMMON DATA(2040),DAT(8000)
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
     1,SLOPE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKD,MVMEAN,
     2TIMSAM,NWDS,IFREQ,MSFC,NFILE,RMS(12),VMEAN(12),NORUN,EXPNO,NOPAGE,
     4AVGMIN(12),AVGMAX(12),NBLK,NSTART,
     3COMTIM
      IA=0
      NSKIP=IBGN
      NSET=(IEND-IBGN+1)/NSCANS
      IC=0
      NTEST3=NSET
      DO 10 NHOR=1,NSET
      NB=(NHOR-1)*NSCANS+NSKIP
      NE=NB+NSCANS-1
   60 CONTINUE
      DO 11 NVER=1,NSECT
      CALL REDTPC (LTN,1,IERR,NW,NUP,DAT)
      GO TO (50,51,50,50,51),IERR
   51 GO TO 30
   50 CONTINUE
C     IC=IC+1
C     IF (IC .GT. IEND) GO TO 998
      DO 14 NPK=NB,NE
      J=0
      NFRM=NPK*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      DO 15 I=NFRM,NTO
      J=J+1
      NTEST1=DAT(I)
      IF (NTEST1 .EQ. NTEST2) DAT(I)=RMS(J)
   15 CONTINUE
   14 CONTINUE
      NEE=NE-1
      DO 16 NPK=NB,NEE
      IA=IA+1
      TIME=TIME+DELT
      DATA(IA)=TIME
      NFRM=NPK*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      DO 17 I=NFRM,NTO
      IA=IA+1
      IF (NDERV) 18,19,18
   18 CONTINUE
      IV=I+NBEAMS
      DATA(IA)=DAT(IV)-DAT(I)
      GO TO 17
   19 DATA(IA)=DAT(I)
   17 CONTINUE
      IF ( IA .LT. NWDS ) GO TO 16
      WRITE (10) (DATA(II),II=1,NWDS)
      WRITE (6,100) (DATA(II),II=1,NWDS)
  100 FORMAT (1X,F9.2,12F9.5)
      IA=0
```

```
 16   CONTINUE
      IA=IA+1
      TIME=TIME+DELT
      DATA(IA)=TIME
      NFRM=NE*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      J=0
      DO 20 I=NFRM,NTO
      J=J+1
      IA=IA+1
      IF (NDERV) 41,40,41
 40   DATA(IA)=DAT(I)
      GO TO 20
 41   DATA(IA)=RMS(J)-DAT(I)
 20   CONTINUE
 11   CONTINUE
      GO TO 60
 30   CONTINUE
      CALL BSFILE (LTN,2)
      CALL SKRBIN (LTN,1,NOP)
      GO TO 10
 999  CONTINUE
      CALL BSFILE (LTN,1)
      CALL SKRBIN (LTN,1,NOP)
 10   CONTINUE
 998  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE XHOR (NSCANS,NSECT,TIME,NSKIP,NTEST2,NSET,DELT,
     1NUP,NDERV,NBCK,IEND,IBGN)
      COMMON DATA(2040),DAT(8000)
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12)
     1,SLOPE(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKD,MVMEAN,
     2TIMSAM,NWDS,IFREQ,MSFC,NFILE,RMS(12),VMEAN(12),NORUN,EXPNO,NOPAGE,
     4AVGMIN(12),AVGMAX(12),NBLK,NSTART,
     3COMTIM
      IA=0
      NSKIP=IBGN
      NSET=(IEND-IBGN+1)/NSCANS
      IC=0
      NTEST3=NSET
    1 CONTINUE
      DO 10 NHOR=1,NSET
   11 CONTINUE
      NB=(NHOR-1)*NSCANS+NSKIP
      NE=NB+NSCANS-1
      DO 12 NVER=1,NSECT
      CALL REDTPC (LTN,1,IERR,NW,NUP,DAT)
      GO TO (50,51,50,50,51),IERR
   51 GO TO 999
   50 CONTINUE
C     IC=IC+1
C     IF (IC .GT. IEND) GO TO 999
      DO 14 NPK=NB,NE
      J=0
      NFRM=NPK*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      DO 13 I=NFRM,NTO
      J=J+1
      NTEST1=DAT(I)
      IF (NTEST1 .EQ. NTEST2) DAT(I)=RMS(J)
   13 CONTINUE
   14 CONTINUE
      NEE=NE-1
      J=0
      DO 15 NPK=NB,NEE
      IA=IA+1
      TIME=TIME+DELT
      DATA(IA)=TIME
      NFRM=NPK*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      IDC=0
      DO 16 I=NFRM,NTO
      IA=IA+1
      IDC=IDC+1
      IF (NDERV) 20,21,20
   20 CONTINUE
      IV=I+NBEAMS
      DATA(IA)=DAT(IV)-DAT(I)
      GO TO 16
C21   DATA(IA)=DAT(I)
   21 DATA(IA)=DAT(I)-RMS(IDC)
   16 CONTINUE
```

```
      IF (IA .LT. NWDS) GO TO 15
      WRITE (10) (DATA(II),II=1,NWDS)
      WRITE (6,100) (DATA(II),II=1,NWDS)
100   FORMAT (1X,F9.2,12F9.5)
      IA=0
15    CONTINUE
      IA=IA+1
      TIME=TIME+DELT
      DATA(IA)=TIME
      NFRM=NE*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      J=0
      IDC=0
      DO 30 I=NFRM,NTO
      J=J+1
      IA=IA+1
      IDC=IDC+1
      IF (NDERV) 41,40,41
C 40  DATA(IA)=DAT(I)
 40   DATA(IA)=DAT(I)-RMS(IDC)
      GO TO 30
 41   DATA(IA)=RMS(J)-DAT(I)
 30   CONTINUE
 12   CONTINUE
      IF (NHOR .EQ. NTEST3) GO TO 10
      CALL BSRECD (LTN,NBCK ,NOF)
 10   CONTINUE
      GO TO 1
999   CONTINUE
      CALL BSFILE (LTN,2)
      CALL SKRBIN (LTN,1,NOP)
      RETURN
      END
```

SUM

```
      SUBROUTINE SUMER (NVAR,DAT,NSECT,NSPAN,ICI,NBEAMS,NFLAG,NBUTT,
     1NBCKUP,LTN,IRT,MOVING)
      DIMENSION SUM(12,200),DAT(1),NDAT(12)
      IF (NFLAG .NE. 1) GO TO 11
      DO 10 I=1,NVAR
      DO 10 II=1,NBEAMS
      SUM(II,I)=0.0
   10 CONTINUE
   11 CONTINUE
      IAA=0
      IA=0
      DO 21 II=1,NVAR
      IAA=IAA+NBEAMS
      DO 22 IB=1,NSPAN
      DO 23 IC=1,NBEAMS
      IA=IA+1
      SUM(IC,II)=SUM(IC,II)+DAT(IA)
   23 CONTINUE
   22 CONTINUE
      IF (MOVING .NE. 0) IA=IAA
   21 CONTINUE
      IF (NFLAG .GE. NSECT) GO TO 30
      NFLAG=NFLAG+1
      RETURN
   30 IA=0
      IF (MOVING .EQ. 0) GO TO 50
      CALL BSRECD (LTN,NBCKUP,NOP)
      ICI=ICI-NSECT+1
   50 CONTINUE
      DO 40 II=1,NVAR
      DO 40 IC=1,NBEAMS
      IA=IA+1
      DAT(IA)=SUM(IC,II)/FLOAT(NSPAN*NSECT)
   40 CONTINUE
      WRITE (11) (DAT(K),K=1,IA)
      NFLAG=1
      NODD=(ICI/2)*2+1-ICI
      DO 200 ICC=1,NVAR
      IF (NBUTT .EQ. 0) GO TO 800
      IF (NODD) 801,800,801
  800 NFRM=ICC*NBEAMS-NBEAMS+1
      GO TO 799
  801 NFRM=(NVAR  -ICC)*NBEAMS+1
  799 NTO =NFRM+NBEAMS-1
      J=0
      DO 201 I=NFRM,NTO
      J=J+1
      NDAT(J)=DAT(I)*1000.0
  201 CONTINUE
C     WRITE (7,1010) ICI,ICC,(NDAT(II),II=1,NBEAMS)
      WRITE (6,1010) ICI,ICC,(NDAT(II),II=1,NBEAMS)
 1010 FORMAT (1X,2I4,12I5)
  200 CONTINUE
      WRITE (6,1011)
 1011 FORMAT (1H ,16H5X5 MOVING MEAN ,/)
      RETURN
      END
```

```
      SUBROUTINE SUHERE (NCOUNT,          NOWAY,NUP,NSIZE,
     1NCHAN,NTEST2,RCDID,IRT)
      COMMON DATA(2040),DAT(8000)
      COMMON START(5),STOP(5),SC(53),VALMIN(12),VALMAX(12),
     1SLOP(12),BIAS(12),LTN,NBEAMS,NFLSKC,NFLSKD,MVMEAN,TIMSAM,
     2NWDS,IFREQ,MSFC,NFILE,RMS(12),VMEAN(12),NORUN,EXPNO,NOPAGE,
     3AVGMIN(12),AVGMAX(12),NBLK,NSTART,COMTIM
      DIMENSION NCHAN(1)
C     CALCULATE INDEXING PARAMETERS
      NREF2=NCOUNT*NBEAMS
      NREF3=NOWAY*NBEAMS
      NUP=NCOUNT*NBEAMS
      NOWAY1=NOWAY+1
      NR=2*NOWAY*(NUP/NSIZE+1)
      NEND=NCOUNT-NOWAY
      NR1=(NOWAY-1)*(NUP/NSIZE+1)
      MN=0
 2006 CONTINUE
      DO 2000 I=1,NCOUNT
      DATA(I)=0.0
 2000 CONTINUE
C     INPUT DAT AND ORGANIZE
      NUPP=1
      DO 2004 I=1,3
      CALL REDTPC (IRT,1,IERR,NW,NUP,DAT(NUPP))
      GO TO (50,51,50,50,51),IERR
   51 GO TO 999
   50 CONTINUE
      IF (I .EQ. 3 ) GO TO 2005
      IF (NR1 .EQ. 0) GO TO 2005
      CALL SKRBIN (IRT,NR1,NOP)
 2005 CONTINUE
      NUPP=NUPP+NUP
 2004 CONTINUE
      MN=MN+1
C     POSITION TAPE FOR NEXT READ
      CALL BSRECD (IRT,NR,NOP)
      DO 2001 I=NOWAY,NEND
      II=1
      DO 2002 K=1,NBEAMS
      NREF1=(I+NCOUNT)*NBEAMS+K
      NSAMP1=NREF1-NREF2
      NSAMP2=NREF1-NREF3
      NSAMP3=NREF1+NREF3
      NSAMP4=NREF1+NREF2
      NTEST1=DAT(NREF1)
      IF (NTEST1 .EQ. NTEST2)DAT(NREF1)=RMS(K)
      NTEST1=DAT(NSAMP1)
      IF (NTEST1 .EQ. NTEST2)DAT(NSAMP1)=RMS(K)
      NTEST1=DAT(NSAMP2)
      IF (NTEST1 .EQ. NTEST2)DAT(NSAMP2)=RMS(K)
      NTEST1=DAT(NSAMP3)
      IF (NTEST1 .EQ. NTEST2)DAT(NSAMP3)=RMS(K)
      NTEST1=DAT(NSAMP4)
      IF (NTEST1 .EQ. NTEST2)DAT(NSAMP4)=RMS(K)
```

```
      P=(DAT(NREF1)-DAT(NSAMP1))**2
      Q=(DAT(NREF1)-DAT(NSAMP2))**2
      R=(DAT(NREF1)-DAT(NSAMP3))**2
      S=(DAT(NREF1)-DAT(NSAMP4))**2
      IF (K .NE. NCHAN(II)) GO TO 2007
      II=II+1
      DATA(I)=DATA(I)+P+Q+R+S
 2007 CONTINUE
 2002 CONTINUE
 2001 CONTINUE
C     WRITE (6,2003) (DATA(I),I=NOWAY1,NEND)
 2003 FORMAT (1X,12F9.5)
      DO 2009 I=NOWAY1,NEND
      WRITE (7,2008) RCDID,MN,I,DATA(I)
 2009 CONTINUE
 2008 FORMAT (1X,A6,2I6,F12.6)
      GO TO 2006
 999  CONTINUE
      CALL BSFILE (IRT,2)
      IF (IRT .EQ. 9 ) CALL SKRBIN(IRT,1,NOP)
      RETURN
      END
```

```
      SUBROUTINE PROB (DAT,NBEAMS,TABLE,DATAB,NPROB1,NPROB2,NPROB3)
      DIMENSION DAT(1),TABLE(101,12),DATAB(1)
      IA=NPROB3
      DO 39 IB=NPROB1,NPROB2
      ID=0
      DO 40 IC=1,NBEAMS
      ID=ID+1
      IA=IA+1
      DO 41 JJ=1,101
      IF (DAT(IA) .GT. DATAB(JJ)+.0175) GO TO 41
      IF (DAT(IA) .LT. DATAB(JJ)-.0175) GO TO 41
      TABLE(JJ,ID)=TABLE(JJ,ID)+1.0
      GO TO 40
41    CONTINUE
40    CONTINUE
39    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PROBJT (DAT,NBEAMS,NBLK,IERR)
      DIMENSION DAT(1),NP(120,120),ALPHA(120),ALPNUM(11)
      DIMENSION NTABL(10),NPSUMX(120),NPSUMY(120)
      DIMENSION CORDX(120)
      NAMELIST/SCAL/XMAX,XMIN,IMX,IMY,RESOL
      DATA NFLAG/0/
      IF (NFLAG .NE. 0) GO TO 1
      READ (5,SCAL)
      WRITE (6,SCAL)
      READ (5,1004) (ALPNUM(I),I=1,11)
      WRITE (6,1004) (ALPNUM(I),I=1,11)
      READ (5,1005) (NTABL(I),I=1,10)
 1005 FORMAT (10I5)
      NFLAG=1
      NRESOL=RESOL
      X=RESOL/(XMAX-XMIN)
      DO 10 I=1,NRESOL
      DO 10 J=1,NRESOL
      NP(I,J)=0
   10 CONTINUE
    1 CONTINUE
      GO TO (50,51,50,50,51),IERR
   51 GO TO 999
   50 CONTINUE
      NCT=NCT+1
      DO 100 III=1,NBLK
      IM=(NBEAMS+1)*(III-1)
      IMM=IM+IMX+1
      DATA=DAT(IMM)-XMIN
      IX=X*DATA
      IF (IX .GT. NRESOL) IX=NRESOL
      IF (IX .LT. 1 ) IX=1
      IMM=IM+IMY+1
      DATA=DAT(IMM)-XMIN
      JY=X*DATA
      IF (JY .GT. NRESOL) JY=NRESOL
      IF (JY .LT. 1) JY=1
      NP(IX,JY)=NP(IX,JY)+1
  100 CONTINUE
      RETURN
  999 CONTINUE
      NCOUNT=(NCT-1)*NBLK
      DO 200 IX=1,NRESOL
      DO 201 JY=1,NRESOL
      PERCNT=(FLOAT(NP(IX,JY))/FLOAT(NCOUNT))*100.0
C     WRITE(6,1000) IX,JY,PERCNT,NP(IX,JY)
  201 CONTINUE
  200 CONTINUE
 1000 FORMAT (3X,I3,1H,,I3,2X,F8.4,5X,I6  )
      DO 300 I=1,NRESOL
      NPSUMX(I)=0
      DO 301 J=1,NRESOL
      NPSUMX(I)=NPSUMX(I)+NP(I,J)
  301 CONTINUE
  300 CONTINUE
```

```
      NTEMP=NP(1,1)
      DO 510 I=1,NRESOL
      DO 510 J=1,NRESOL
      IF ( NP(I,J) .GT. NTEMP) NTEMP= NP(I,J)
  510 CONTINUE
      NFACT=NTEMP/6
      WRITE (6,1006)
 1006 FORMAT (1H1)
      DO 500 IYY=1,NRESOL
      CORDX(IYY)=FLOAT(IYY)/X
      IY=NRESOL+1-IYY
      CORDY=FLOAT(IY)/X
      II=0
      DO 505 IX=1,NRESOL
      II=II+1
      DO 506 I=1,8
      IF (NP(IX,IY) .GE. NTABL(I)) GO TO 506
      ALPHA(II)=ALPNUM(I-1)
      GO TO 505
  506 CONTINUE
  505 CONTINUE
      IF (IY .NE. NRESOL)  GO TO 512
      WRITE (6,1008) CORDY,(ALPHA(I),I=1,NRESOL)
 1008 FORMAT (1X,F6.3,4X,120A1)
      GO TO 513
  512 CONTINUE
      IF (IY .NE. NRESOL/2) GO TO 514
      WRITE (6,1008) CORDY,(ALPHA(I),I=1,NRESOL)
      GO TO 513
  514 CONTINUE
      IF (IY .NE. 1) GO TO 511
      WRITE (6,1008) CORDY,(ALPHA(I),I=1,NRESOL)
      GO TO 513
  511 CONTINUE
      WRITE (6,1007) (ALPHA(I),I=1,NRESOL)
 1007 FORMAT (9X,1H-,120A1)
  513 CONTINUE
  500 CONTINUE
      WRITE (6,1009) CORDX(1),CORDX(60),CORDX(120)
 1009 FORMAT (10X,F6.3,53X,F6.3,52X,F6.3)
      WRITE(6,1001)
 1001 FORMAT(34H1  P(I,J)       P(0/0)        P(N)     )
 1004 FORMAT (1H ,120A1)
      DO 400 J=1,NRESOL
      NPSUMY(J)=0
      DO 401 I=1,NRESOL
      NPSUMY(J)=NPSUMY(J)+NP(I,J)
  401 CONTINUE
  400 CONTINUE
      WRITE (6,1003)
      WRITE(6,1002) (NPSUMX(IJ),NPSUMY(IJ),IJ=1,NRESOL)
 1003 FORMAT (24H1   P(X)          P(Y)     )
 1002 FORMAT (3X,I6,6X,I6    )
      RETURN
      END
```

```
      PLOTT

      DIMENSION RCDID(15),D1(222),D2(222),D3(222),D4(222),D5(222)
      DIMENSION D6(222),D7(222),D8(222),D9(222),D10(222),D11(222)
      DIMENSION D12(222),NPLOT(12),TIME1(222),DAT(5000),NWHICH(5)
      DIMENSION XARRAY(12),YARRAY(12),NSCANS(120)
      NAMELIST /NAMDAT/ NBEAMS,NWHICH,NFILE,LTN,NPLOT,NSCANS,NSIZE,NO
      EQUIVALENCE (RCDID(13),NCOUNT)
994   FORMAT (1X,I6)
1007  FORMAT (12A6)
1008  FORMAT (1X,I6)
      DO 99 I=1,120
      NSCANS=0
99    CONTINUE
      XNORM=1.0
      DO 100 II=1,12
 100  NPLOT(II)=0
      READ (5,NAMDAT)
      WRITE (6,NAMDAT)
      DELT=.1
      TIME = -DELT
      NCT=0
      CALL CAMRAV(9)
      READ (5,1007) (XARRAY(I),I=1,12)
      READ (5,1007) (YARRAY(I),I=1,12)
      DO 998 IAA=1,NFILE
      REWIND LTN
      IAB=NWHICH(IAA)-1
      IF (IAB)11,11,12
12    CONTINUE
      DO 10 IAC=1,IAB
      CALL SKFBIN (LTN,1,RD)
      WRITE (6,994) IAC
10    CONTINUE
11    CONTINUE
      READ (LTN) (RCDID(I),I=1,15)
      CALL FLD (NCT,19,17,1,RCDID(13))
      NCOUNT=NCT
      NNUP=NCOUNT*2
      NUP=NCOUNT*NBEAMS
      ISU1=0
60    CONTINUE
      ISU1=ISU1+1
      IF (ISU1 .GT. NO) GO TO 999
      NUPER=NSCANS(ISU1+1)-NSCANS(ISU1)-1
      WRITE (6,994) NUPER
      NUPER=NUPER*(NUP/NSIZE+1)
      WRITE (6,994) NUPER
      IF (NUPER .LE. 0) GO TO 997
      DO 996 ISU2=1,NUPER
      CALL SKRBIN(LTN,1,NUP)
996   CONTINUE
997   CONTINUE
      WRITE (6,995) NSCANS(ISU1+1)
995   FORMAT (1X,13HSCAN NUMBER=    ,I6)
      CALL REDTPC(LTN,1,IERR,NW,NUP,DAT)
      WRITE (6,1008) NW
```

```
      GO TO (50,51,50,50,51),IERR
51    GO TO 999
50    CONTINUE
      I=1
      DO 80 ICC=1,NCOUNT
      NFRM=ICC*NBEAMS-NBEAMS+1
      NTO=NFRM+NBEAMS-1
      TIME=TIME+DELT
      TIME1(ICC)=TIME
      D1(ICC)=DAT(I)/XNORM
      I=I+1
      D2(ICC)=DAT(I)/XNORM
      I=I+1
      D3(ICC)=DAT(I)/XNORM
      I=I+1
      D4(ICC)=DAT(I)/XNORM
      I=I+1
      D5(ICC)=DAT(I)/XNORM
      I=I+1
      D6(ICC)=DAT(I)/XNORM
      I=I+1
      D7(ICC)=DAT(I)/XNORM
      I=I+1
      D8(ICC)=DAT(I)/XNORM
      I=I+1
      D9(ICC)=DAT(I)/XNORM
      I=I+1
      D10(ICC)=DAT(I)/XNORM
      I=I+1
      D11(ICC)=DAT(I)/XNORM
      I=I+1
      D12(ICC)=DAT(I)/XNORM
      I=I+1
80    CONTINUE
      J=1
      IF (NPLOT(J) .NE. 1) GO TO 101
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D1 )
      CALL PRINTV (4,4HCH 1,150,1010)
      CALL PRINTV (90,RCDID,212,1010)
      J=J+1
101   IF (NPLOT(J) .NE. 2) GO TO 102
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D2 )
      CALL PRINTV (4,4HCH 2,150,1010)
      CALL PRINTV (90,RCDID,212,1010)
      J=J+1
102   IF (NPLOT(J) .NE. 3) GO TO 103
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D3 )
      CALL PRINTV (4,4HCH 3,150,1010)
      CALL PRINTV (90,RCDID,212,1010)
      J=J+1
103   IF (NPLOT(J) .NE. 4) GO TO 104
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D4 )
```

```
      PLOTT

      CALL PRINTV (4,4HCH 4,150,1010)
      CALL PRINTV (90,RCDID,212,1010)
      J=J+1
104   IF (NPLOT(J) .NE. 5) GO TO 105
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D5 )
      CALL PRINTV (4,4HCH 5,150,1010)
      CALL PRINTV (90,RCDID,212,1010)
      J=J+1
105   IF (NPLOT(J) .NE. 6) GO TO 106
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D6 )
      CALL PRINTV (90,RCDID,212,1010)
      CALL PRINTV (4,4HCH 6,150,1010)
      J=J+1
106   IF (NPLOT(J) .NE. 7) GO TO 107
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D7 )
      CALL PRINTV (90,RCDID,212,1010)
      CALL PRINTV (4,4HCH 7,150,1010)
      J=J+1
107   IF (NPLOT(J) .NE. 8) GO TO 108
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D8 )
      CALL PRINTV (90,RCDID,212,1010)
      CALL PRINTV (4,4HCH 8,150,1010)
      J=J+1
108   IF (NPLOT(J) .NE. 9) GO TO 109
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D9 )
      CALL PRINTV (4,4HCH 9,150,1010)
      CALL PRINTV (90,RCDID,212,1010)
      J=J+1
109   IF (NPLOT(J) .NE. 10) GO TO 110
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D10)
      CALL PRINTV (90,RCDID,212,1010)
      CALL PRINTV (4,4HCH10,150,1010)
      J=J+1
110   IF (NPLOT(J) .NE. 11) GO TO 111
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D11)
      CALL PRINTV (90,RCDID,212,1010)
      CALL PRINTV (4,4HCH11,150,1010)
      J=J+1
111   IF (NPLOT(J) .NE. 12) GO TO 112
      CALL QUIK3L(-1,TIME1(1),TIME1(NCOUNT),0.,255.,1H.,XARRAY,
     1YARRAY,-NCOUNT,TIME1,D12)
      CALL PRINTV (90,RCDID,212,1010)
      CALL PRINTV (4,4HCH12,150,1010)
      J=J+1
112   CONTINUE
      GO TO 60
999   CONTINUE
998   CONTINUE
      CALL CLEAN
      STOP
      END
```

DISTRIBUTION

DIR
DEP-T
A&TS-PAT
PM-PR-M, Mr. Goldston
A&TS-MS-H
A&TS-MS-IP
A&TS-MS-IL (8)
A&TS-MS-TU, Mr. Wiggins (6)
A&TS-PAT, Mr. Wofford
AD-S, Dr. Stuhlinger

PD-DIR, Dr. Lucas
PD-SS-MGR, Mr. Brooksbank
PD-MP-T, Mr. Craft
PD-MP-DIR, Mr. Downey

AST-U, Col. Mohlere

S&E-DIR, Mr. Weidner

S&E-R
- Dr. Johnson
- Mr. Miles
- Mr. Lake
- Mr. Chase
- Mr. Coons

S&E-COMP
- Mr. Houston
- Mr. Jones
- Mr. Cochran
- Dr. Hoelzer
- Dr. Kerner
- Mr. Cozelos

S&E-PT, Mr. Angele

S&E-ASTR
- Mr. Lucas
- Mr. Zimmerman

S&E-CSE, Mr. Fredrick

S&E-ASTN-PTD, Mr. Hopson

PM
- Mr. Kurtz
- Mr. Peddycoart
- Mr. Belew

S&E-SSL
- Mr. Heller
- Dr. Sieber
- Mr. Snoddy

S&E-AERO
- Dr. Geissler
- Mr. Horn
- Mr. Dahm
- Mr. W. Vaughan
- Mr. Baker
- Mr. Lovingood
- Mr. Lindberg
- Mr. Turner
- Dr. F. Krause
- Mr. Heaman
- Mr. Huffaker
- Mr. Ellner (25)
- Mr. Kadrmas
- Mr. K. Johnston
- Mr. Funk
- Mr. Cummings
- Mr. Murphree
- Mr. Jandebeur
- Mr. Bush
- Dr. Grosche
- Mr. Bilbro
- Mr. Wilhold
- Mr. Jayroe
- Dr. DeVries
- Mr. Cliff
- Mr. Stephens
- Mr. Dalton

RSIC (2 copies)

Scientific & Tech Info Facility (25)
P.O. Box 33
College Park, Md 20740
ATTN: NASA Rep. (S-AK/RKT)

NASA-Langley Research Center
Langley Field
Hampton, Virginia 23365
ATTN: Tech Library (2)

NASA-Goddard Space Flight Center
Greenbelt, Md. 20771
ATTN: Tech Library (2)

NASA-Ames Research Center
Moffett Field
Mountain View, Calif. 94035
ATTN: Tech Library (2)

NASA-Lewis Research Center
21000 Brookpark Rd.
Cleveland, Ohio 44135
ATTN: Tech Library (2)

Jet Propulsion Lab.
Calif. Inst. of Tech.
4800 Oak Grove Dr.
Pasadena, Calif. 91103
ATTN: Tech Library (2)

NASA-Kennedy SpaceCenter
Kennedy Space Center, Fla. 32899
ATTN: Tech Library (2)

IITRI
10 W. 35th St.
Chicago, Illinois 60616
ATTN: Col. Ferrell
Mr. Phillips
Mr. Weigandt
Mr. Betz

Northrop-EMD
Technology Drive
Huntsville, Ala 35805
ATTN: Mr. Grady
Mr. Hand
Dr. Su
Mr. Pooley

ESSA-Wave Propagation Lab
National Bureau of Standards Bldg.
Boulder, Colorado 80302
ATTN: Dr. Beam
Dr. Derr
Dr. Strauch

Alabama A&M
Huntsville, Alabama
ATTN: Dr. Foster
Mr. C. Foster

Mr. T. A. George
Environmental Protection Agency
Water Quality Office
Washington, D. C. 20242

Colorado State University
Ft. Collins, Colorado 80521
ATTN: Prof. Sandborn
Dr. Marshner
Dr. Sadeh
Dr. Ruff
Prof. Reiter
Dr. Simon
Dr. Winn
Dr. Rasmussen

University of Oklahoma, Dept of Physics
Norman Oklahoma 73069
ATTN: Dr. Canfield
Dr. Fowler
Dr. Day
Mr. Lysobey
Dr. Corvin

FAA, NO-10
800 Independence Aven, S.W.
Washington, D.C. 20590
ATTN: Dr. K. Powers
Dr. J. Powers

NASA Headquarters
Dr. von Braun, AAD
Mr. Foster, AAD-ID
Mr. Ginter, RF
Mr. Badgley, SM
Mr. Spreen, SFM
Mr. Tepper, SRD
Mr. Vacca, MLT
Mr. DeMerritt, RVI
Mr. Rosche, RV-2
Mr. Reiffel, MA
Mr. Underwood, RVA
Mr. Liningston, MTG
Mr. Spriggs, -SA
Mr. Koutsandreas, SRB
Mr. Centers, SRF
Dr. Park, SRR
Mr. Gessow, RR-2
Mr. East, RET
Mr. Gilchrist, RFS
Dr. Summers, SRF
Lt Col Lehman, SRF
Mr. McGowan, RAO
Mr. Michel, RWS
Mr. Parkinson, RAA
Mr. Rekos, RAP
Mr. Schwartz, RRF
Mr. Harper, RD-A
Dr. Kurzweg, RR
Mr. Peil, MH
Mr. Cerreta, RV-1

MSFC—RSA, Ala