012268

LARS Information Note ~~02268~~

Purdue University

LABORATORY FOR AGRICULTURAL REMOTE SENSING (LARS)

INTERFEROMETER DATA PROCESSING SYSTEM

Paul E. Anuta

Preface:

The system described in this manual consists of several hardware/ software items constructed to meet the needs of LARS researchers using spectrometers built by the Block Engineering Company. Its development took place during most of 1967, and it was designed specifically for the model 195T, 195E, and P4 instruments which were on hand at that time. If different instruments are used, changes in the system may be required. The initial work on the interferogram processing problem was carried out by Terry Phillips and Barry Merrill under the direction of Dr. David Landgrebe. The hardware development was carried out by John Clevenger and Robert Smith under the direction of Dr. Roger Holmes. The software system was developed, programmed, and documented by Paul E. Anuta of the LARS Data Handling Staff.

TABLE OF CONTENTS

Illustrations

LARS INTERFEROMETER DATA PROCESSING SYSTEM

I. Introduction

Data produced by the LARS interferometer instrument group must be digitized, Fourier transformed, and reformatted before it can be used by the research staff. The system described in this manual carries out all the required operations on the interferogram data to produce graphs and a storage file of the spectrums contained in the interferograms. The two basic parts of the system are the digitizing process which is built around the LARS tape-to-tape system, and the data processing program. Interferogram data is brought to the A/D system via analog tape where it is digitized and written on standard computer tape. The digitized data is then read, averaged, Fourier transformed, and plotted by the processing programs on the IBM 360/44 computer and a data storage file tape is written.

The digitization portion of this report consists of specifications and procedures for converting the data from a loosely formatted analog form to a uniform digital format on what is called a bulk digital data tape. This process is time consuming and requires considerable operator intervention to identify and tag the data. The discussion and directions herein are intended to aid the operator in accomplishing this task and to speed up the process. The data processing part consists mainly of a description of the program which converts the bulk data to spectrum plots and file entries which contain all information necessary for interpretation of the data. The programs combine many functions which are controlled by input cards and keyboard responses. The data can be averaged, Fourier transformed, plotted, printed, graphed, punched out, and written on an output tape under control of the operator. The system requirements are first discussed briefly then the parts of the system are discussed in detail.

II. Data Characteristics and System Requirements

1. Field Spectrometer Characteristics

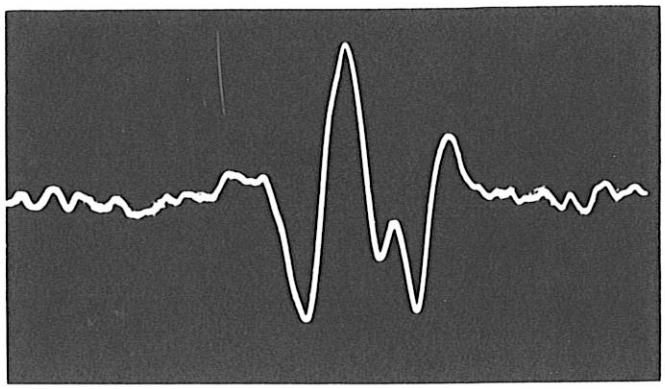
The Block Manufacturing Company interferometers which are used in the spectral radiance studies produce data which is not directly usable by the researchers. The output of the instruments is a sequence of short waveforms ranging from .1 to 1 sec. in duration, each of which contains all the spectral information received from the source. Each signal is the Fourier transform of the actual spectrum observed. A large number of these signals is produced for any one subject and the number of observations of various subjects is also large. Thus, an interferogram data handling problem exists which the LARS data handling group has the responsibility to solve. This section details the problems and describes the structure of the interferometer data system which was developed.

The data which the interferometers produce is generally in a form of a sinx/x waveform. One of these waveshapes is produced for each scan of the mirror in the 195T and 195E instruments or of the polarizing crystal in the P4-model. A picture of a typical output waveform appears in Figure 1a. The duration of the output signal for each scan from the instruments on hand is:

195T and 195E - .125 to 1 sec.

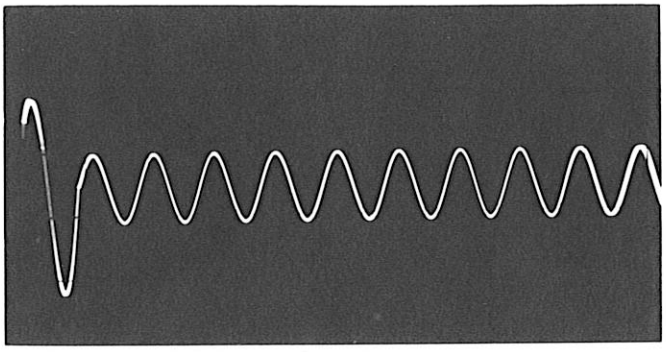
P4 - .1 sec.

The exact values are available from calibration data.



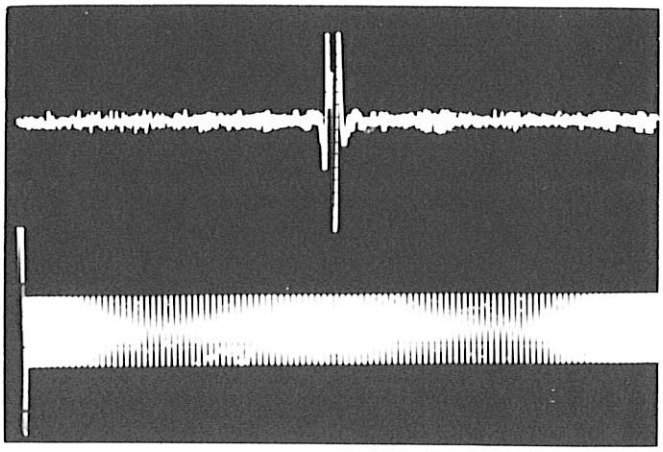
100 Millisec

a. Interferogram Waveform



3 Millisec

b. Clock Waveform



1/4 to 1 Sec.

c. Clock and Interferogram Waveform over full scan

Figure 1 Interferogram Waveforms

The instruments each have a clock output signal which is required for conversion of the data waveform to digital form. The clock from the 195T unit is an oscillator generated nominal 2KC sine wave which is on only during the period the data waveform is being produced. The 195E clock is approximately 8 KC. The clock from the P4 unit is produced by a light source, moire' grating, and photocell configuration which produces a distorted sine wave which varies in frequency about a nominal 10KC. The first cycle of the clock is approximately twice the magnitude of all the rest. The clock signal shapes are shown in Figure 1b. Both signals for a full scan appear in Fig. 1c. The data and clock signals are recorded on an analog tape recorder along with control and identification information and delivered to the data handling group. The responsibility of the D. H. group begins at this point.

2. Fourier Transformation Requirement. As noted above, the data waveform is related to the spectrum observed by an integral transformation. The waveform is expressed as:

$$f(t) = \frac{1}{2}\int_0^\infty I(\nu)\tau(\nu)d\nu + \frac{1}{4}\int_0^\infty \left[I(\nu)\tau(\nu)\exp(i\phi(\nu))\right]\exp(i2\pi\frac{L\nu}{T})d\nu$$

where: $f(t)$ - output from instrument $o \leq t \leq T$

T - scan time of instrument (sec)

$I(\nu)$ - spectrum intensity (watts/cm^2 - sterad - cm^{-1})

$\tau(\nu)$ - overall instrument transmittance

$\phi(\nu)$ - Phase shift between direct and reflected waves due to instrument effects. (radians)

L - total displacement of mirror (microns)

ν - Wave number ($\nu = .01/\lambda$)(cm^{-1})
(wave number is the number of cycles which would be observed in 1 cm.)

In order to obtain $I(\nu)\tau(\nu)$ the inverse Fourier transform with respect to the variable t must be taken.

$$Y(\nu) = I(\nu)\tau(\nu) = \int_{-\infty}^{\infty} f_1(t)\exp(i2\pi \frac{L\nu t}{T})dt$$

where $$f_1(t) = 4f(t) - 2\int_{-\infty}^{\infty} I(\nu)\tau(\nu)d\nu \qquad i = \sqrt{-1}$$

Computation of $Y(\nu)$ by a digital computer requires that $f_1(t)$ be sampled and represented by a digital number at each sample point. Sampling at twice the maximum frequency expected in the interferogram will give sufficient information for reconstruction of the waveform. The clock signal is used for sampling the interferogram. Its frequency is selected by the manufacturer to give a sufficient sampling rate. The $f_1(t)$ is then available in the computer as a sequence

$\{F_i\}$ $i = -\frac{N}{2} + 1$ to $i = \frac{N}{2}$ which is to be Fourier transformed. (N is the total number of samples taken of the interferogram.)

3. Data Smoothing. The interferogram contains noise components that must be suppressed in some manner. The noise is primarily due to local variations in the source environment and instrument mechanical and electrical noise. Several means of smoothing are available such as analog and digital filtering, and averaging data from many interferograms. In the system described in this manual only data averaging is carried out. In this approach the digital samples from many scans are averaged in the computer and the Fourier transform is taken of the averaged set.

4. Calibration and Reformatting Requirement

Output data is required to be in a spectrum amplitude versus wavelength form. The real frequency value of the transform output must be determined and the data manipulated to produce the required wavelength points. Instrument transfer calibration is not carried out by this system. This operation must be carried out by researchers using the data.

III. System Structure

The interferometer data handling system consists of four parts as indicated in the flow diagram in Figure 2. The analog to digital conversion process is carried out on the LARS A/D system. The data processing part is carried out by a computer program written for the LARS computer, the 3rd part is a filing program which maintains and accesses a tape containing all the interferometer data. The fourth part is the CALCOMP plot facility. CALCOMP plots can be delivered to the researchers after an expected delay time of two to three days after receipt of the analog data tape, depending on the availability of technical personnel and operational state of the equipment. Line printer plots are immediately available from the IG file. The parts of the system are discussed in detail below.

1. Digitization Process. The recording process during field data collection does not lend itself to precise formatting of the data. A recorder is used which is not synchronized with the instrument and no automatic tagging capability exists for identifying what is put on the analog tape. Therefore, considerable human intervention is required to record the interferometer data and variability in format is the consequence. A procedure has been worked out for recording and converting data to digital form which minimizes lost time. The operator identifies the run by sequence number, source details, instrument settings, and other comments which are written on a data identification sheet which accompanys the data tape.

The recorded data is delivered to the data handling system via a data link from the truck containing the field recorder. A van operator and an A/D system operator are required for digitizing. The A/D operator sequences through the runs on tape digitizing and recording the data on a seven track digital recorder. The system which carries out this process is shown in Figure 3. The A/D operator has as control inputs a voice channel and a clock

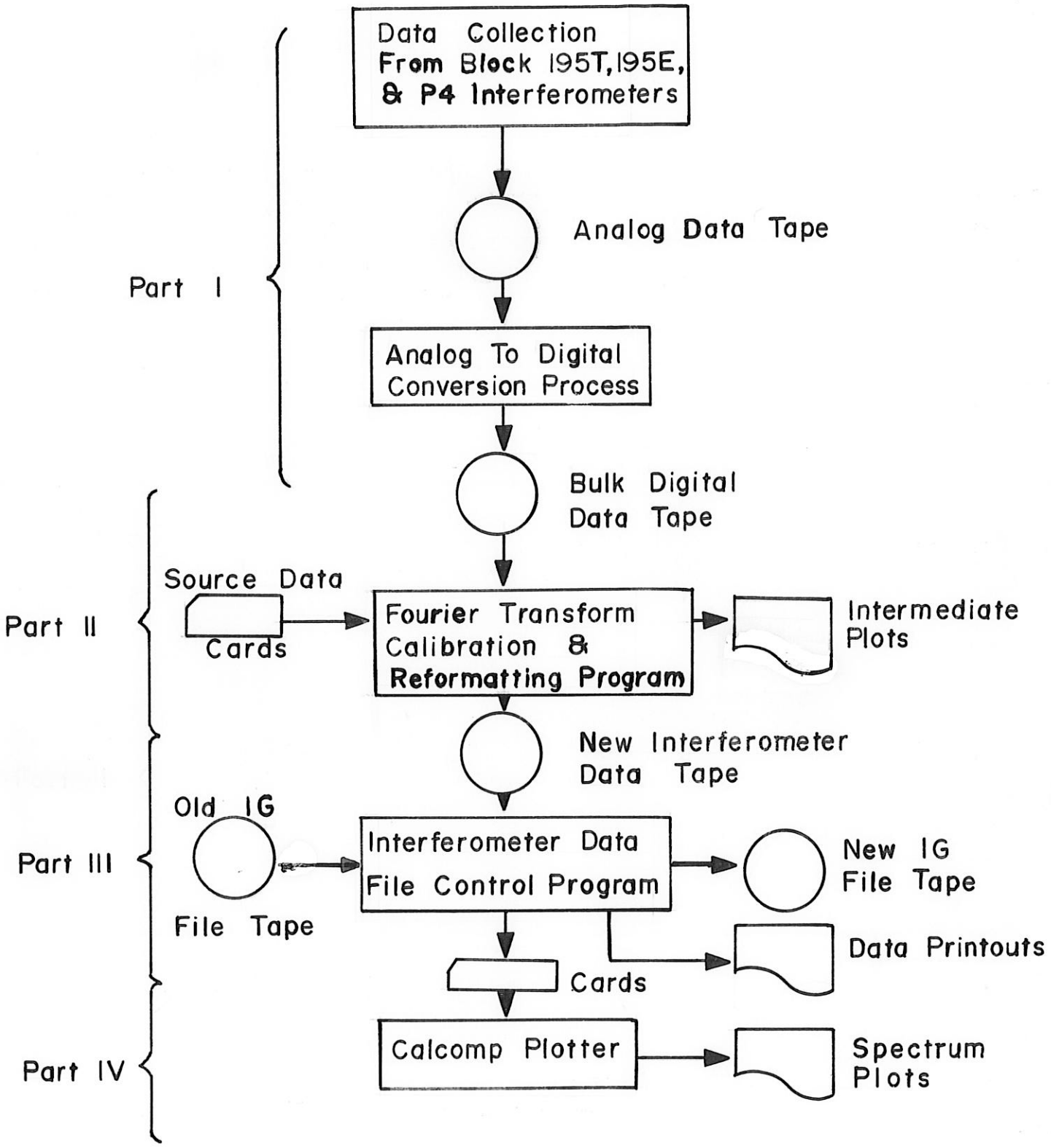


Figure 2. Interferogram Processing System

signal. The voice channel from the van operator gives information on the next run on tape and the starting and stopping of the clock signal delimits the interferogram run. A 12-digit ID number is assembled by the A/D operator which identifies the run and this is written on the digital tape before the data. The A/D conversion process is then begun and continues until the clock signal from the analog recorder stops. The operator repeats the procedure for each run on the tape. The result is a digital data tape containing samples of the interferometer output signal.

The format used in the digitization process is dictated by the core memory capacity of the IARS digital computer. Due to memory limitations, the size of any one data block which can be written on tape is restricted to three hundred 32-bit words. Inputs from some instruments have as many as 4160 samples per scan which would require 2080 words of core space to read the tape record, thus the data must be recorded in seven separate digital tape records. Also, the number of samples received from the instruments varies. These factors led to the decision to write the data on tape in a "free" format and design the computer processing program to scan the tape records and extract complete interferograms. In order to implement this approach, it was necessary to digitize and record the clock signal along with the data samples. The first clock cycle has twice the normal amplitude and therefore can be used to mark the start of the interferogram. The characteristics of the clock and data signals were pictured in Figure 1. Also, the Model P4 interferometer has two channels of output which must be simultaneously recorded whereas the Model 195 instruments have one. The A/D converter has the capability of digitizing only an even number of channels. Thus for the P4, a four-channel conversion is required.. Table 1 contains the data characteristics pertinent to the A/D conversion process.

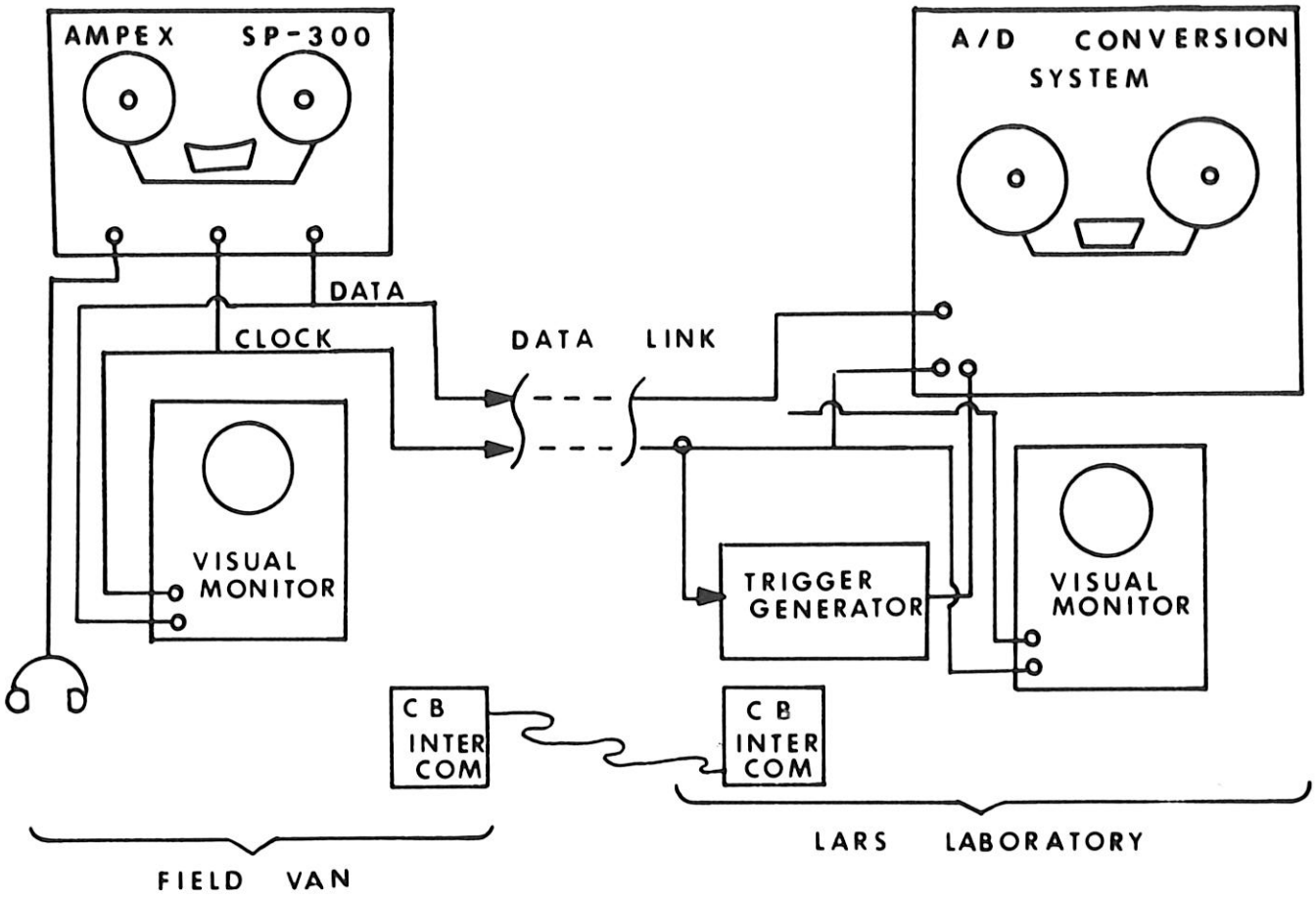


Figure 3. LARS Data Link

Table 1 - Digital Data Tape Characteristics

Instrument	Band	Data Channels	Samp/Scan	Digital Channels	Samp/Tape Record
195T	2 to 16μ	1	251 to 2052	2	582
195E	2 to 6.5μ	1	251 to 4160	2	582
P4	.35 to 2.5μ	2	948	4	291

The data is digitized to eight bits. This gives a resolution of 1 part in 256, or .4%, which is considered to be adequate since the instrument accuracies do not approach this value. The digital data tapes are delivered to part two of the system where the data processing is carried out.

The clock signal obtained from the instrument is not of the proper form for driving the IARS A/D system trigger circuits. Also, it is desired that the peak of the first cycle of the clock be sampled. These requirements led to the development of a special trigger generator circuit which is a part of the digitizing section of the system. This element amplifies the clock signal from the analog tape and produces a trigger pulse for every negative going zero crossing of the clock signal plus a trigger pulse at the peak of the negative swing of the first clock cycle in a scan. This circuit is discussed in detail in Appendix I.

2. Data Processing Program. The digitized interferometer data is processed by a three-phase computer program written for the IARS System 360 Model 44 computer. The first phase reads the interferogram data from the digital tape which was recorded in the free format discussed above and reassembles complete interferograms. It then averages a requested number of the interferograms and the second phase Fourier transforms the resultant smoothed interferogram. The third phase is called into core which calibrates, reformats, and plots the spectrum data.

a. Data Retrieval & Averaging

The magnitude of the first clock cycle is recorded along with the interferogram data in the A/D conversion process. This sample marks the start of an interferogram. The data averaging part of the processing program searches for these start flags and begins collection of a complete interferogram in memory at this point. If a tape record is exhausted before the next start flag is reached, the next tape record is read and the process continues. When the next flag is reached the interferogram is complete and accumulation of the next can begin. The interferograms are averaged by adding the values of the samples read from the tape to the ones already in core. After a specified number have been averaged the interferogram is fed to the Fourier transform program. The effect of averaging on the interferogram is pictured in Figure 4.

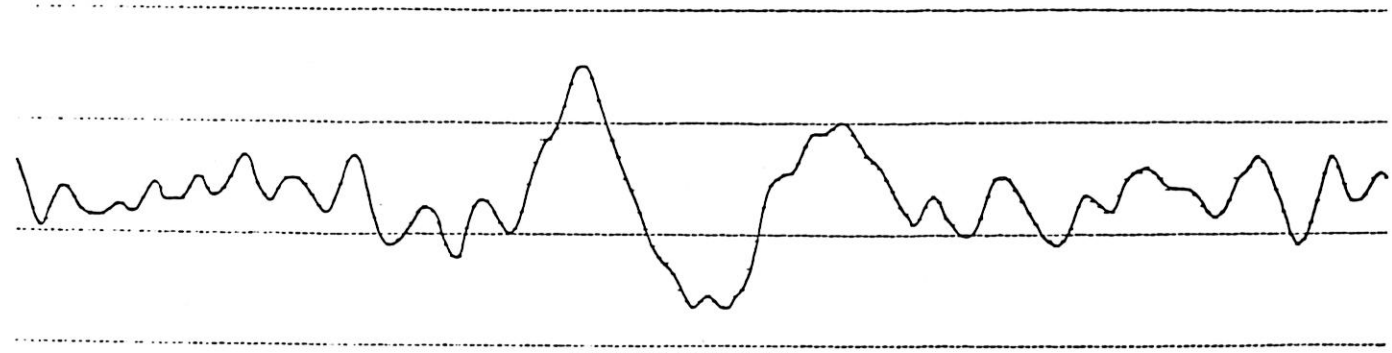
b. Fourier Transform Process

The transform program computes the discrete Fourier Transform of the input data points. It uses an algorithm developed by Cooley and Tukey (see Appendix II) which increases the computation speed for the transform by a factor of LOG_2N/N compared to step by step integration techniques. (N is the number of points transformed.) The problem is to compute the values of a_k in the Fourier series:

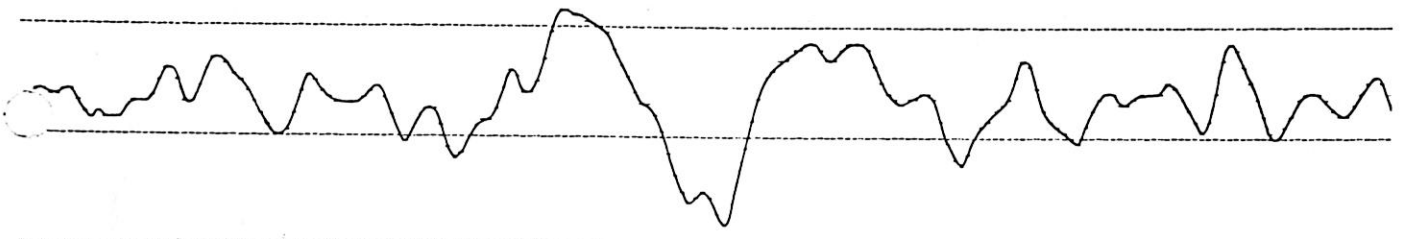
$$x_j = \sum_{k=o}^{N-1} a_k \exp(i2\pi jk/N) \qquad (i = \sqrt{-1})$$

Where the x_j are the values of the N input data points (j = 1, 2, ..., N-1) and the vector $\{a_k\}$ represents the spectrum of frequencies contained in the data. The formula for the a_k is the dual of the one for the series:

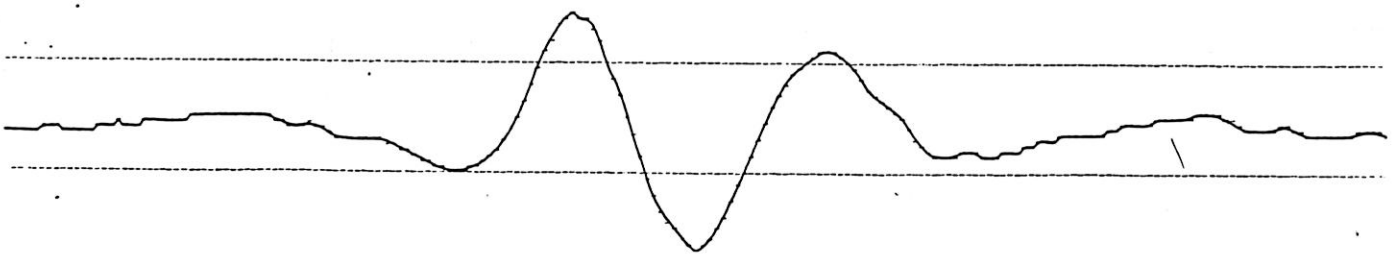
$$a_k = \frac{1}{N} \sum_{j=o}^{N-1} x_j \exp\left(\frac{-i2\pi jk}{N}\right)$$



INTERFEROGRAM PRODUCED BY ONE
SCAN OF INTERFEROMETER MIRROR
RUN NO.= 1102387001
SOURCE BLACK BODY
INSTRUMENT BLOCK 1957
SCAN NO 2



INTERFEROGRAM PRODUCED BY ONE
SCAN OF INTERFEROMETER MIRROR
RUN NO.= 1102387001
SOURCE: BLACK BODY
INSTRUMENT: BLOCK 1957
SCAN NO.4



INTERFEROGRAM PRODUCED BY AVERAGING
73 SCANS OF INTERFEROMETER MIRROR
RUN NO.= 1102387001
SOURCE: BLACK BODY
INSTRUMENT: BLOCK 1957

Figure 4. Effect of Averaging Interferogram

The algorithm uses a precomputed sine and cosine table and a bit by bit computation of the Fourier coefficients. The program accepts N data points and produces a transform containing N/2 points. Due to storage limitations in the computer a maximum of 1024 points can be accepted. Thus, the output spectrums contain 512 or less points. Each point produced represents the radiation energy at a specific visual/IR band frequency. The program assumes a fundamental period of 1 sec., and a_1 is the fundamental component, a_2 the first harmonic, etc., and a_o is the d. c. value of the x vector. The real frequency value of the fundamental component is $1/\tau$ where τ is the time duration spanned by the N data points transformed. If f_s is the frequency at which the data was sampled then the sample interval is $1/f_s$ and $\tau = N/f_s$. Thus, the real frequency value of a_1 is:

$$f_1 = \frac{1}{\tau} = \frac{f_s}{N}$$

and the real frequency value for any a_i is then

$$f_i = \frac{fs}{N} i \qquad i = 1, 2, \ldots, N/2$$

where: N = number of data points transformed.

This frequency is in the audio band since the signal represents an interference pattern caused by a slowly moving mirror or polarizing crystal depending on instrument type. The instrument output frequencies are related to the input visual/IR band frequencies by the geometry and dynamics of the instrument.

The 195T and 195E instruments use the Michelson interferometer principle which uses a moving mirror that causes intensity variations at the detector window having an audio band rate. Assume a single frequency visual/IR band input signal with wavelength λ microns and let T be the sweeptime of the mirror in sec. and L the sweep length in microns. A full cycle of intensity variation will pass a given point of the detector field of view for each $\lambda/2$ displacement of the mirror. Thus, the frequency of the signal seen by the detector will be:

$$f_d = 1/t_\lambda$$

$$t_\lambda = t_m * \lambda/2$$

$$t_m = \frac{T}{L}$$

or $$f_d = \frac{2L}{T\lambda}$$

Where: t_λ is the time for one pattern to pass a point at the detector (sec.)

t_m is the time for the mirror to move one micron (sec.)

f_d is the audio band output freq. (cps) from the detector

λ is the visual/IR band input wavelength (microns)

T is the sweep time of the mirror (sec.)

L is the sweep length of the mirror (microns)

Thus, the audio band output from the Fourier Transform can be related to actual input if T and L are accurately known for the instrument. For the P4 polarization interferometer the geometric structure is such that it is easier to make a direct calibration to get the scale factor rather than compute it. The scale factor for the 195 instruments is computed as follows:

We have: $$f_i = \frac{f_s}{N} i$$

$$f_s = \frac{M}{T}$$

and $$f_d = \frac{2L}{T\lambda}$$

Where M is the number of points obtained from one scan of the mirror.

Combining and solving for λ

$$\lambda = \frac{2L(N)}{T(\frac{M}{T})i}$$

$$= \frac{2LN}{Mi}$$

The identity $f_i = f_d$ is implied.

Thus, the scale factor multiplies the inverse of the sequence number of the spectrum point:

$$\lambda(i) = \frac{K_s}{i}$$

$$K_s = \frac{2LN}{M}$$

where L = sweep length in microns

N = number of samples transformed

M = number of samples from one full scan of the mirror.

i = spectrum data point number (i.e. i = 0 is the d.c. value, i = 1 is the fundamental, etc.)

$\lambda(i)$ = wavelength of ith point in microns

K_s = scale factor in microns

Example: Assume T4, L4 settings which have calibrated values of:

T = 1 sec.

L = 221 microns

These settings result in nominally 2052 samples out indicating a 2.052KC clock. A typical transform will use 1024 of these points.

Thus:

$$K_s = \frac{2*221*1024}{2052} = 221 \text{ microns.}$$

The difference between N and M is due to two factors. (M is the number of points obtained from the instrument and N is the number used in the transform, $N \leq M$.) First, most of the information carried in the interferogram is in the region around the center of the scan. To improve the signal to noise ratio, the data centered around the center 1024 points of the interferogram is transformed. Secondly, for certain settings of sweep time and sweep length, over 4000 data points are produced in one scan. A limit of 1024 points for any interferogram exists for this system; thus, only the center 1024 of the 4000 are available to be transformed. The number of spectrum

points produced by the program is half the number fed in since the output spectrum is periodic over the number of points which was transformed then the second half of the output data is a mirror image of the first half. Also, the transform routine operates on 2^k points (k a positive integer) so the number of points obtained will always be 512 or 256, etc.

Part two of the program carries out the Fourier transform process and feeds the spectrum information in terms of relative frequency (i.e. frequency number i in the above development) to part three of the program. The transform values are in terms of amplitude and phase angle. The transform vector consists of up to 512 sequential data point pairs, the first of each pair being magnitude and the second the phase angle in degrees. The phase angle is normalized by subtracting out the linearly varying component; thus, the values obtained are the deviations from a linear curve. Appendix II presents additional discussion of the Fast Fourier transform in the form of IARS Information Note 61367 and the SHARE abstract of the transform program.

c. Calibration & Reformatting

The averaging and transform programs both fill all of the core space available, thus a third program phase was written which carries out the remainder of the tasks. The conversion of the frequency axis to wavelength is first carried out by this program, using the scale factor developed above. The relationship between frequency and wavelength is hyperbolic and a one to one mapping of the frequency points is not possible since uniformly spaced samples are desired in wavelength. Thus, interpolation must be carried out to produce uniform points in λ. The interpolation formula used is:

$$V_\lambda = V(\text{ENTIER}(\frac{K_s}{\lambda})) + \left[V(\text{ENTIER}(\frac{K_s}{\lambda}) + 1) \right.$$

$$\left. - V(\text{ENTIER}(\frac{K_s}{\lambda}) \right] * (\frac{K_s}{\lambda} - \text{ENTIER}(\frac{K_s}{\lambda}))$$

where: V_λ is the amplitude value of the spectrum at wavelength λ.

K_s is the scale factor.

V(i) is the amplitude of the spectrum at frequency i, where i is an integer.

ENTIER (x) is the greatest integer less than or equal to x.

This is a two step hyperbolic-linear interpolation and the results produced appear satisfactory.

After the data has been reformated, it is plotted in two ways. A line printer graph is made which is of good quality; however, it contains only the discrete data points -- the curves must be drawn by hand. A continuous line plot is obtained from a CALCOMP plotter which gives an immediately usable output. Figure 5 shows a typical spectrum from the printer with curve drawn in. Figure 6 is a CALCOMP plot of similar data. The final operation of the IG processing program is to write a complete interferogram data set on tape. The format of the tape is defined in section 4. This tape can be used directly by researchers if desired or its contents can be added to the file tape discussed next.

3. File Control Program

A large number of interferograms and associated spectrum and source data will be accumulated as research proceeds, and it became evident during development of the system that some file building program would be required. Thus, a filing procedure and program is included in the system. The procedure specifies that the existing file tape cannot be written on during editing. When a file is being changed, a new tape must be created which contains the updated file. This procedure prevents accidental destruction of the existing file due to machine or program error. After editing is complete, the new file can be checked for accuracy and the old file can be saved or destroyed as desired. The file program is also to be used for reading data from the file. Use of the program is discussed in Section IV. The structure of the program is discussed below.

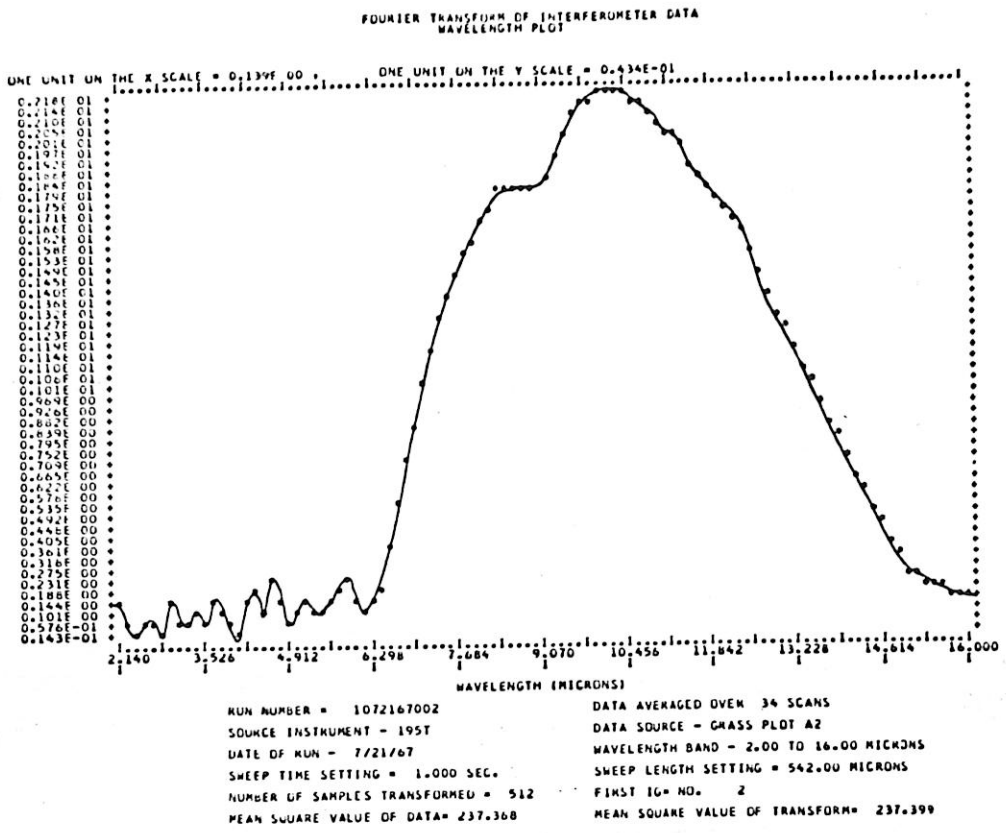


FIGURE 5. LINE PRINTER PLOT

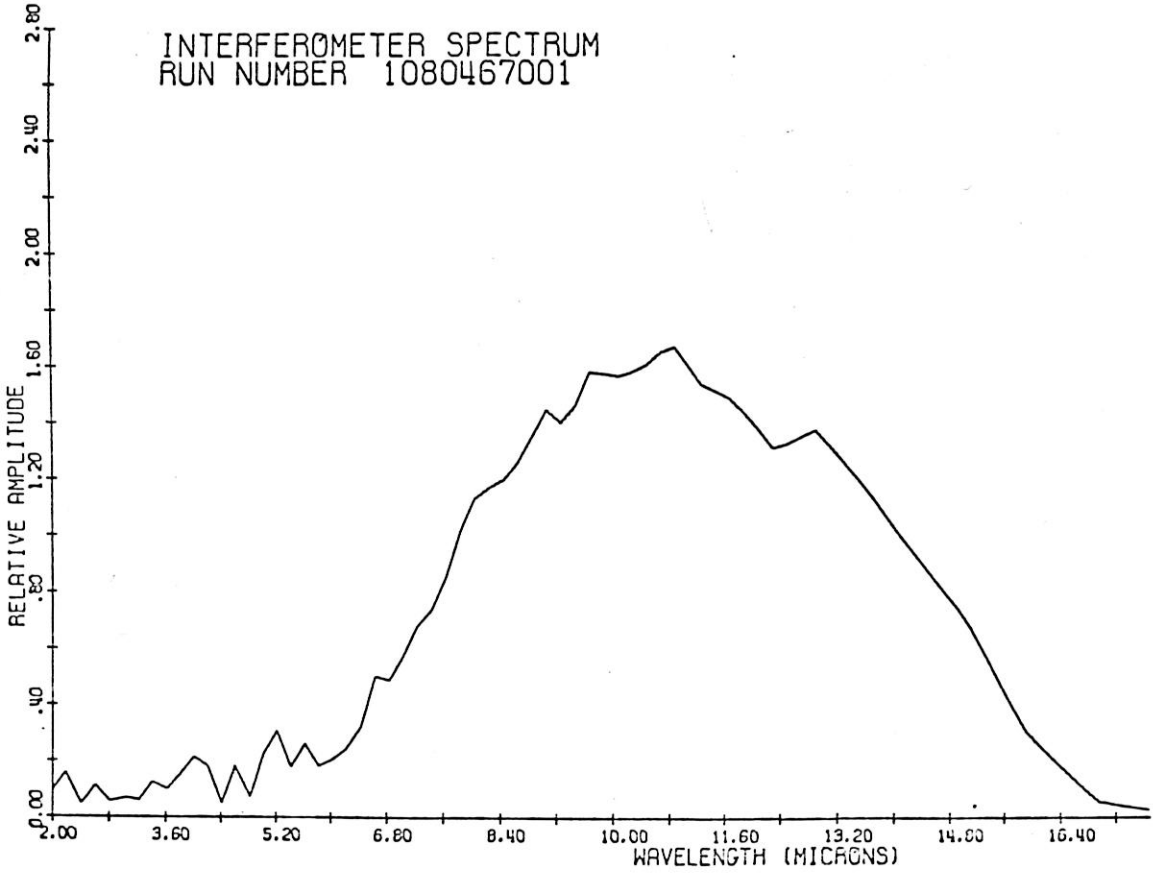


FIGURE 6. CALCOMP PLOT

The file control program consists of a main program and two overlayed phases containing graphing and misc. subroutines. The main program performs three basic functions: Table of contents reading, Data reading, and File editing. The file tape format is defined in section 4. The table of contents resides at the head of the tape and contains ID No. and source data label for each interferogram on the tape. The read optionallows printed or punched output of data in the file. The edit optionallows new interferogram data to be added to the file and data to be deleted.

The IG file tape consists of two regions; the "Table of Contents" region and the "Data" region. The new IG data tape consists of only a data region. In the add mode the file control program builds a table of contents on the new file from the TOC on the old file and the new IG data on the new IG tape. It then constructs a new data region from the data on the old file and the new IG tape. Interferogram data is ordered in the file according to the date the data was taken, the type of instrument, and the order of the run on any one day. The delete option simply causes the specified IG's to be removed from the file. A combination optionis also available which replaces specified IG's on the old file with new IG's with the same number from the new IG tape. This option is included so that data can be updated in one step instead of two. Whenever a new file is formed edit operations on it can be performed only after it has been removed from unit 180 and mounted on unit 181. Dataor table of contents read operations can be performed from any of the three tapes whenever desired via keyboard request. A system data flow and tape unit diagram is presented in Figure 7. A complete data processing and file updating operation can be performed at one time using all four of the LARS tape units.

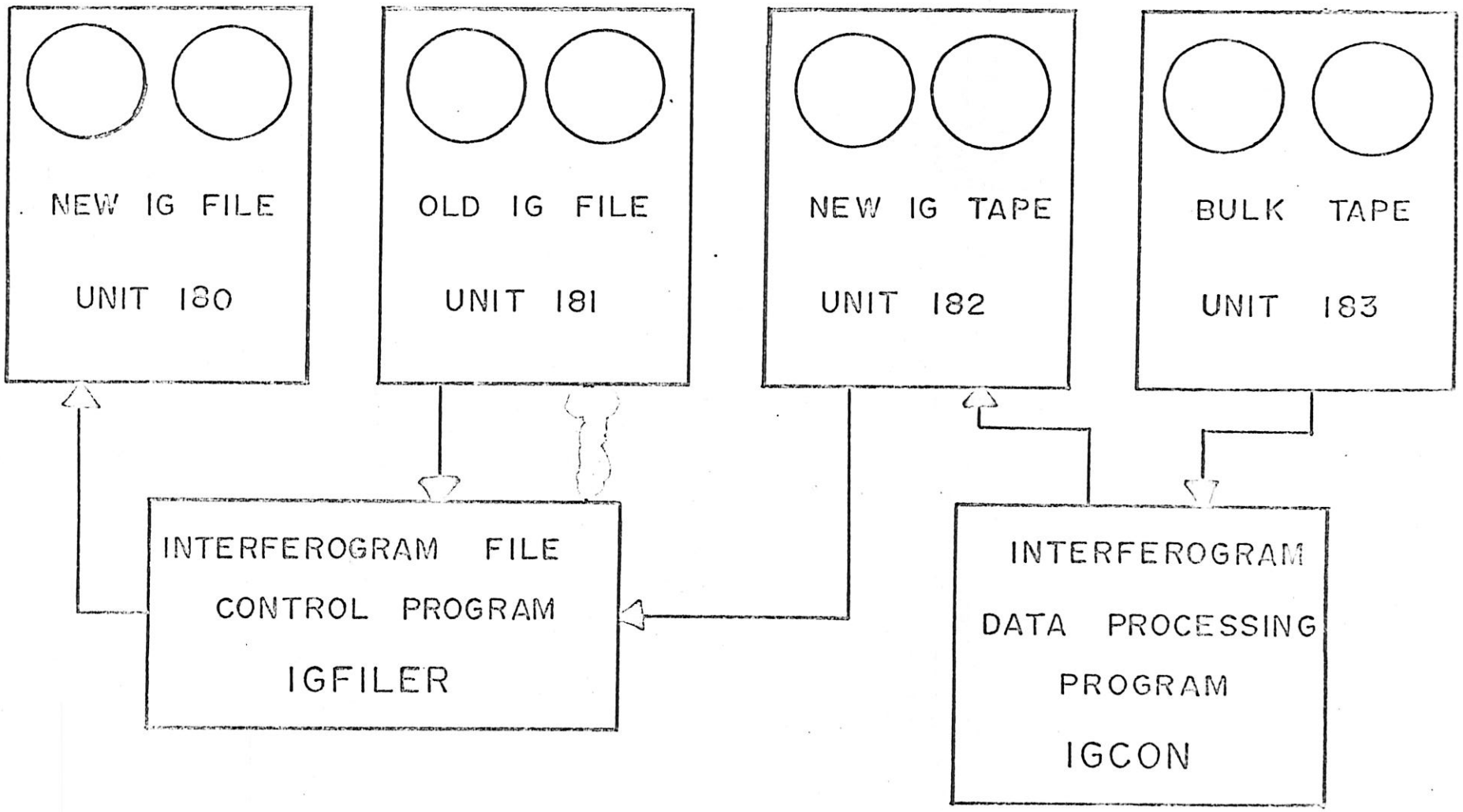


FIGURE 7. IG SYSTEM DATA FLOW

4. Data Tape Formats

a. Bulk Data Tape Format

The interferogram data is digitized and recorded on tape in variable length format which will allow a maximum of flexibility in the A/D process. The only restriction which must be imposed is maximum record length which can be written. Due to computer memory limitation no more than 1200 bytes can be read from any one tape record. An interferogram can, however, continue from one tape record to the next. A formal definition of the bulk IG data tape is as follows:

1. The Bulk Tape consists of a sequence of "runs" with an end-of-file mark between each run and two end-of-file marks at the end of all runs on a tape. A run is an ID Record followed by a sequence of Data Records. An ID record consists of 72 bit number to identify the run. A data record is a sequence of data words from one or two channels of input, plus the clock channel.

2. The ID Record is 12 six-bit integers having the following meaning (left to right):

Digit

1 - Identifies the unit from which data was taken.
- 1 = Block 195T
- 2 = Block 195E
- 3 = Block P4

2,3 - Month the data was taken.

4,5 - Day the data was taken.

6,7 - Year the data was taken.

8 - Setting of the sweep time control (1 thru 4)

9 - Setting of the sweep length control (1 thru 4)

10,11,12 - Run serial identification number.

3. The Data Records have 2 channels of data for the 195T and E and 4 channels for the P4. Each data quantity is an 8-bit sample from the sensor or clock. The format is:

a. 195T and 195E

Channel 1 = Data
Channel 2 = Clock

b. P4

Channel 1 = Data from silicon sensor (.35 to 1 micron)
Channel 2 = Data from PbS sensor (1 to 2.5 microns)
Channel 3 = Clock
Channel 4 = Blank (Not used)

(The blank channel is a result of the incapability of the A/D system to write an odd number of channels.) The processing program does not store the blank channel of course and no core is wasted because of this. Tape space is wasted but the data volume is not expected to be a problem for the bulk interferogram tape. Interferograms continue from one tape record to the next and are marked by the clock channel.

b. Interferogram File Tape Format

The Fourier Transformed calibrated, and reformatted interferometer data is written on a file tape for permanent storage. The IG file system uses several tapes in it's operation but the data format is the same for all tapes containing IG's which have been processed by the Fourier transform program. Four items of data are stored for each interferogram: identification, graph coordinates, averaged interferogram, and transformed interferogram. The first two items constitute the user output data and the second two are backup records of the actual data points operated on by IGCON. These are retained for future research and for verification of the graphic output. Also a fifth distinguishable tape record which marks the end of data for the tape is also defined. A formal definition of these records follows. All data is written with FORTRAN commands, thus the records referred to below are "logical records" and may consist of many physical tape records.

1. IG Data File Tape consists of a "Table of Contents" and a sequence of "Interferogram Data Sets" with the end of all data on tape marked by an "End of Data" record. An Interferogram Data Set consists of an "ID Record" followed by the "Graph Coordinate Record", the "Averaged Interferometer Waveform Record", and the "Transformed Waveform Record". This format is the same for all IG's.

2. ID Record consists of fifty full words (200 bytes) containing all the identification data for the data which follows it in the Interferogram Data Set. The meaning of each word is given below:

I. Integer Format Section

Word No.	Contents
1	Sortable Identification Number
2	User Identification Number
3	Month Data Taken
4	Day Data Taken
5	Year Data Taken
6	Number of Data Points in Spectrum (Each point has magnitude and phase value, i.e. 2 words per point)
7	Data Point having maximum magnitude
8	Unit code (1=195T, 2=195E, 3=P4)
9	P4 channel (ch1=.35-1μ,ch2=1-2.5μ)
10	Number of scans averaged in this transform
11	Number of first scan which was used
12	Number of data points in averaged interferogram
13	Number of data points expected in interferogram
14	Number of points in graph of spectrum
15	0

II. Floating Point Section

Word No.	Contents
16	Sweep time } for 195T and E only
17	Sweep Length } for 195T and E only
18	Lower limit of inst. response (microns)
19	Upper limit of inst. response (microns)
20	Mean square value of the interferogram
21	Mean square value of the transform
22	0
23	Scale factor used to produce spectrum plot
24	0
25	0

III. Alphanumeric Section

26	Blanks
27	Blanks
28	Instrument Name
29-48	Source Data and Comments

IV. Integer Termination Section

49	0
50	999999

3. Graph Coordinate Record consists of x, y pairs of points for the calibrated and reformatted spectrum graph. There are nominally 100 points in this record (200 words) although more may be present in some cases. The number of points in this record is always given in ID word 14.

4. Averaged Interferogram Waveform Record consists of the 1024 or less points obtained by averaging many interferograms. It is the actual data used by the Fourier transform routine. It is in integer format and the data has values from 0 to 255. The record is always 1024 words long. The number of data points in the record is given by ID(12).

5. Transformed Waveform Record contains the 512 data points obtained from the Fourier transform rountine which used the data defined above.

Each point consists of a relative magnitude and a phase angle (in degrees) value. The record always consists of 1024 words. The number of point pairs in the transform is given by ID(12)/2.

6. End of Data record consists of 50 words the first 49 of which are zeros and word fifty is 666,666. (ID (50) = 666666)

7. Table of Contents consists of a sequence of 22 word records containing ID numbers and source description for all interferometer data in the tape. It's format is:

Word 1 = Sortable ID number

Word 2 = User ID number

Word 3-21 = Alphanumeric source description words

Word 22 = 999999

There are as many 22-word table of contents records as there are interferogram data sets on the tape. The table of contents proceeds all interferogram data sets on the tape. The end of the table of contents is marked by an "End of Table" record which is 22 words in length which contain:

Words 1-21 = 0

Words 22 = 333333

IV. User Procedures

The following sections describe the steps necessary to process interferometer data written on analog tape.

1. Data Link Setup

Interferometer data is transmitted to the IARS analog to digital converter via a three channel cable link which runs from the IARS field van to the A/D system. The IARS field van must be parked within cable reach of the terminal mounted at the bottom of the center rear window of the IARS building. A general operator familiarity with power up and operating procedures for the field van is assumed in this report. Interferometer data system operators must be trained in the operation of the van prior to use of this system.

Park the van as indicated above and power up the Ampex SP300 recorder, and monitoring scope. Connect the link between the building terminal and the panel above the left cab door. The data link terminates on the right of the "IARS Data Link" panel beneath the recorder.

Data from the Block 195T and E instruments requires 2 channels for transmission to the A/D system. The Block P4 requires 3. Connect the clock and data outputs from the tape recorder to the cable terminals on the panel. The data channels for 195T and E are cable 1 (or 1 and 2 for the P4) and the clock output is connected to cable 2 for the 195T and E or to cable 3 for the P4.

Mount the tape containing data to be processed on the recorder and connect the appropriate channels to the data link. Use the monitor scope to insure that the desired signals are present when the recorder is started.

The recorder can be operated by either a van operator or by the remote control system. A van operator can communicate with the A/D operator via

"walkie talkie" Citizen Band link. If remote control is used, position the tape at the point which will allow the operator to proceed most conveniently.

2. A/D System Setup

Digitization of the interferometer requires that certain connections be made and initialization procedures be carried out before processing can begin. A basic knowledge of how to operate the A/D system is required and this manual does not attempt to supply this information. An IG system operator must be previously trained on the A/D system by the IARS data handling staff.

Follow normal power-up procedure for the A/D system and mount a blank bulk tape RING IN on the CDC 9110 drive. Power up an oscilloscope and the IG SYSTEM TRIGGER GENERATOR. Connections at the A/D system were roughly pictured in Figure 3. A more detailed diagram is presented below in Figure 8. The trigger generator is housed in a bench rack near the A/D System. Cable connections are to be made as indicated in the Figure.

The connections shown are for 195T and 195E type data. P4 data requires the use of channel 3. Data channel gain is controlled by the DANA amplifiers in the middle of the left A/D rack. The amplifiers are connected to BNC connectors on the jack panel below. Amplifiers in the 1, 2 (and for P4 the 3) position are used in the system. The 2 blue colored amplifiers must be plugged into positions 1 and 2. These units are not gain calibrated and can be used as continuously variable gain units by using the adjusting screws on the face of the amps. If P4 data is processed, a third variable gain unit must be obtained by some means.

The dynamic range of the A/D system is $\pm$ 10 volts and, for best digitizing, it is recommended that the data channel gain be set so that most of the dynamic range is used. The level of the clock signal into the trigger generator should be set so that the negative peak of the first clock cycle from the interfero-

NOTE: ALL CABLES RG-58U WITH BNC CONNECTORS

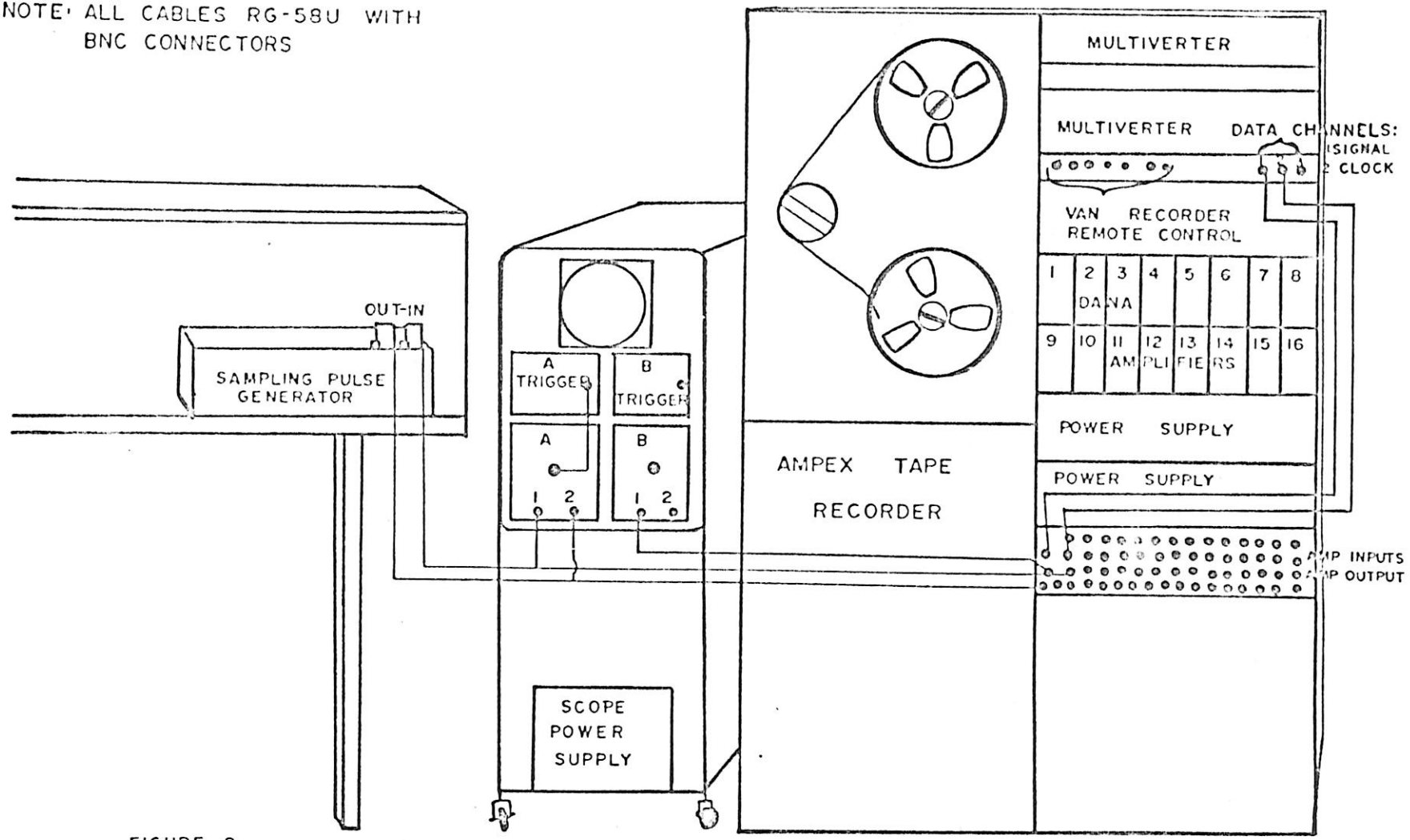


FIGURE 8

gram has a vlue of -10 volts. The trigger generator is set to expect this value. Adjustment of the trigger generator is discussed in Appendix I. When satisfacotry signals are observed, digitizing can begin.

The block length on the A/D system is to be set to 776 for both 195 and P4 data.

Set the "multiverter" channel select switches on the two units to 0 for 195T & E data and to 1 for P4 data. The Bits select switch on both units is to be set to 8. The mode switches are to be set to "normal".

The CDC tape drive is readied by pressing LOAD after mounting a blank tape then pressing REWIND and READY.

The A/D system is ready for digiting after these operations.

3. Digitizing Procedure

Interferometer digitizing is begun by entering a 12-digit ID number on the A/D control panel. This is done by setting in the numbers on the dials and pressing "ENTER". The ID number is composed as follows: (Described on page 21 also.)

XMMDDYYTLNNN

Where:

X = The Instrument Code

1 = 195T
2 = 195E
3 = P4

MMDDYY = Month, Day, Last two digits of year
T = Sweep Time Setting
L = Sweep Length Setting
NNN = Run Serial Number

After the ID is entered, digitizing may begin. The analog tape in the field van is started either by pressing the remote control start switch or by the van operator. When data and clock signals appear on the scope press, the $\frac{\text{START}}{\text{STOP}}$ switch on the A/D control panel. This starts the digitizing process. When the signals stop on the scope press, the $\frac{\text{START}}{\text{STOP}}$ switch again to stop the A/D process.

The end of a run on the digital tape is to be marked by one end of file mark. Press the E.O.F. switch to enter this mark. Digitizing of one run is then complete.

This process is repeated for each run which is to be digitized from the analog tape. When all runs are digitized, two end-of-file marks are to be placed on the digital tape to mark the end of all data on the tape. When finished, press CLEAR and REWIND. Then press REWIND again to unload the tape from the vacuum columns. Remove the tape from the unit and remove the ring. The digital tape is now ready for processing.

The analog recorder in the van and can be controlled by the remote switches on the IARS Data Link panel. These switches correspond to the same ones on the recorder. Their use is the same as if the operator were at the recorder.

4. Interferogram Data Processing Procedure

The interferogram data processing operation is carried out by a computer program written for the IBM system 360 Model 44 computer using the PS44 programming system. The program reads, averages, Fourier transforms, plots, and outputs the interferogram data on tape. The procedure for using this program is detailed in the paragraphs that follow. A complete description of the program is included in Section V.

a. Initial Setup. The processing program is named IGCON and is stored on a magnetic tape labeled IGCON/IGFILER PROGRAM. Obtain this tape and the program load cards necessary to begin the run. If these cannot be found the required cards are:

```
//IGLOAD JOB
//SYSOOO ACCESS SCRTAP, 181=
//SYSIPT ACCESS IGCON, 180='LARSn'
//SYSRDR ACCESS IGCON, 180='LARSn'
/*                                   (n is the tape number)
/&
```

Mount the program tape with ring out on unit 180 and "ready" the unit. Mount a scratch tape on unit 181. These two items constitute all the software input required to process interferograms.

The system input required is the PS44 system disk which must be mounted and readied in the 44 disk drive. The initialization procedure to be followed when starting the disk is given on the front page of the "Current System Status" book which is to be found on the computer console. Initialize when the disk ready light comes on. When the keyboard returns the "Intervention Required OOA" message you are ready to load the above input cards. About 75 seconds are required to load the program after START is pressed on the reader. Execute this process.

The data inputs to process interferograms consist of a tape containing digitized interferograms and data cards. Remove the ring from the bulk data tape to be processed and mount it on unit 183 and "ready" the unit. Punch up two data cards for each interferogram run to be processed from this tape according to the following format: (All numbers must be right justified in their field.)

Card 1: Run Control Card Col. 1-12 ID Number of the run to be processed under control of this card. Must match number on bulk tape.

Col. 13-16 The first spectrogram to be processed in this run. (Must be 2 or greater) If zero or blank "2" will be assumed.

Col. 17-20 The last spectrogram to be processed in this run. If zero or blank all will be processed. (i.e. program proceeds to end of file mark.)

Col. 21-24 The spectrogram interval. Skips N-1 interferograms between each one processed. If zero or blank "1" is assumed. (i.e., 1 means every interferogram will be processed.)

Col. 25-28	Threshold for start clock pulse detection. If zero or blank standard value of 50 is assumed.
Col. 29-32	Start sample for graph and punch of individual or averaged interferogram.
Col. 33-36	Last sample for graph and punch of individual or averaged interferogram.
Col. 37-40	Not Used.
Col. 44	Mode Flag; If = 0 or blank the interferograms called for all averaged. If = 1 each interferogram is processed individually.
Col. 48	If = "1" Fourier transform taken over center half of data if possible. If "0" or blank 1024 points are transformed.
Col. 52	Print Flag; if 0 or blank a numerical printout of the individual or averaged interferogram is produced. If "1" no printout is obtained.

Col. 56 Graph Flag; "1" produces a complete graph of the averaged data. If "0" or blank a truncated graph of the center portion of the averaged interferogram is produced.

Col. 60 Punch Flag; Punches averaged or individual interferogram in I4 format if = 1. Zero or blank = no punch.

Col. 64 Scan Flag Insertion Switch: If = 0 no action. If = 1 scan flags are inserted in data as indicated on scan flag data modification cards. This feature is discussed below.

Col. 68 Tape Write Flag; If = 0 the Fourier transformed data is written on unit 182. If = 1 keyboard action is required to write data. No action if = 2.

Col. 72 If = "1" the transformed, calibrated, and reformatted data is punched out in F8.3 format. This option is used to punch data for the CALCOMP plotter. If = "0" or blank no action.

Col. 76	End of Runs Flag: If = "1" and Col. 68 is "0", 1, or blank an end of runs record is written on the output data tape indicating the end of data on the tape. If = "0" or blank a check code is written in the ID record for the data indicating that the record is a valid but non-terminal ID.
Col. 80	Skip Flag: If = "1" the IG is skipped. If = "0" or blank the IG is processed. If = 2 runs are skipped until the one having a matching ID is found. If the ID on the desired run is incorrect this option will fail. If the skip option is used card No. 2 must be omitted.

Card No. 2: Run Label Card Col. 1-80 Alphabetic-numeric data defining the source of the interferogram and any other comments desired. Any FORTRAN character can be used.

It can be seen from examination of the control card inputs that a standard default option set has been chosen which requires that only one

item be punched in the card for a standard run. If a correct 12-digit ID number is punched and the rest of the card is left blank, the program will average all the interferograms found in the run, Fourier transform the result, and write the resultant spectrum on tape on unit 182. Furthermore, to terminate a job all that must be added is a "one" in column 76 of the last control card to be processed.

The "run label card" must always be present even if blank except if Col. 80 is a "1" on the run control card. (i.e. skip option)

b. Program Operation. To execute IGCON after it has been read from tape onto the disk, the "Execute IGCON" job must be loaded in the card reader.

The IGCON job cards are:

```
//IGCON     JOB
//SYS005    ACCESS   BULKTAP,183='BULKTP'
//SYS004    ACCESS   NEWTAP,182='NEWIG'
//GO    EXEC    IGPHASE1
```

The data cards follow the EXEC card.

When all IGCON processing is finished the program asks for another bulk tape to be mounted. If no more bulk tapes are to be processed, the job is terminated by pressing PSW restart on the computer operator panel. The program remains on the disk and can be re-executed if desired. After PSW restart is pressed, the IGFILER portion can be executed as discussed below. If neither IGCON or IGFILER are to be executed, the program must be deleted from the disk to make the space available for other person's programs. The Delete Job is as follows:

```
//DELETE JOB
//SYSAB2    ACCESS    SDSABS,OCO='SYSRES'
//DELETE    SDSABS(IGPHASE1,IGPHASE2,IGPHASE3,IGFILER,GRAPHER,SUBS)
//CONDENSE  SDSABS
/*
/&
```

When the IGCON program is loaded and executed, the first message which is printed on the keyboard is: "MOUNT BULK DATA TAPE ON 183 ...". When the tape is mounted, enter EOB to proceed.

The first time the computer refers to any tape unit in a job it will ask that the tape be mounted and readied. This requires that a second EOB be entered. If the output tape write standard option is used a program mount message and a system mount message will be produced and two EOB's will have to be entered as above.

The program reads the bulk tape records, assembles complete interferograms from one or more tape records, and averages the required number. During this process several error conditions can occur. There are two types of error--data errors, and transmission errors. The data errors are missing or incorrect scan start flags, signal saturation, noise in the data, or incorrect ID no's. The transmission type error refers to tape read errors or errors in the format of the data such as missing ID or end of file records. The program detects and responds to these errors and produces keyboard messages in some cases. The error list below is a guide to error response.

The program produces three types of line printer output. During reading and averaging a tape record summary line is printed if a scan start flag was found in that record. It gives the data record number in the run, the block count, the sample locations in the record where the scan flags were found and the number of samples found in the interferogram most recently completed. Error messages are printed at the time the error occurs during reading. The second type of output

is a numerical printout and graph of the averaged or individual interferogram which is to be transformed. The third type is a graphic representation of the spectrum.

Three graphs are produced for each interferogram. The first is a plot of the output of the Fourier transform program in terms of relative intensity and relative frequency. The second is a plot of intensity versus wavelength in which frequency coordinates of each data point were converted to wavelength and replotted. This process causes a bunching and spreading of the points due to the hyperbolic transformation involved. The third graph is an interpolated representation in which the points are uniform in wavelength, and the span of points is selected so that the wavelength axis divisions are easily interpreted. The final interpolated graph is written out on tape unit 182 if the standard write option is selected. Under standard option operation the program is completely automatic and requires no operator intervention. The error conditions which can occur are as follows:

c. Error Conditions.

1. Keyboard message: "NO ID RECORD FOR THIS RUN" indicates that no 12 digit ID number was found where expected on the bulk tape. The program sets the tape ID vector to zero and reports that ID numbers do not agree. (See next message for operator action.)

2. Keyboard message: "ID NUMBERS DO NOT AGREE." The option is then given to correct the card ID number which is the one that will be written on the output file. If the card ID is correct entering "NO" to the re-read ID question will cause the program to proceed with processing the data that follows under option control of that card. Options cannot be changed by keyboard action.

3. Keyboard message: "DO YOU WISH TO PROCESS THIS RUN?" allows the operator to process or skip a run for which the ID is wrong or missing.

4. Keyboard Message: "RESTART JOB OR CONTINUE?" Entering "REST" causes the bulk tape to be rewound, a new control and source card to be read and processing restarted. "CONT" causes the present run to be skipped and forward

search on the bulk tape to continue. Keyboard message "READ NEW CONTROL CARD?" is produced when "CONT" is entered. If YES is entered a new control and source card is read when the next EOF mark is reached. If NO is entered the control and source cards last read are retained.

5. Line printer message: "PERMANENT PARITY CHECK ERROR . . ." indicates that a longitudinal and/or vertical parity error exists for a record read from the bulk tape. The data from that record may have errors in it. Cleaning tape drive may eliminate this condition. Inspect output data for abnormal characteristics.

6. Line Printer Message: "READ OPERATION FOR RECORD n WAS TERMINATED ..." indicates that the tape record was longer than expected and reading was terminated at the count requested by the program. Cleaning tape drive may eliminate this condition. It may be due to faulty operation of the A/D system. Output data should be carefully inspected for abnormal characteristics.

7. Keyboard message: "INVALID RESPONSE - RETYPE" indicates keyboard input is unacceptable. Re-enter the correct response.

8. Keyboard message: "2EOF MARKS FOUND AFTER RUN NOT FLAGGED AS LAST ON CONTROL CARD ... ", indicates that column 76 on the last data card was not a "1" but 2 EOF's were found after that run indicating end of data on tape. If end of job is intended enter YES on keyboard. If not, enter NO and processing will continue.

d. Scan Flag Insertion Option

If Col. 64 is 1 on the run control card the program expects scan flag data. If a scan flag is missing for an IG and this IG is badly needed the missing flag can be added to the current tape record in core if desired with this option. The operation requires one card for each tape record to be fixed. Col. 1 thru 4 must contain the tape record number to be modified, Col. 5 thru 8 the number of flags to be inserted in that record, then Col. 9 thru 12 contains the sample point in that record where the first flag is to be inserted, Col. 13 thru 16 the second, etc.

Eight flags can be added to any one tape record if desired.

5. Interferometer Data File Reading and Updating

A multi-function file control program exists as part of the interferometer data system which allows a user to read and edit the tape containing the interferometer data file. The file format was defined in Section III-4. The file contains identification, spectrum, and averaged interferogram data. The READ option is used to extract this information from the file tape. The EDIT option is used to change the contents of the file. The file control program obtains all its input commands from the keyboard, no data cards are required.

a. Program Loading and Execution.

The file control program is stored on the IGCON/IGFILER program tape in absolute form and is read into the computer along with the interferometer data conversion program. Execution of the filing program can begin as soon as a interferogram conversion run has finished or it can be executed as an independent program. The loading procedure is the same as that for the IGCON part described above. To execute the file program load the IGFILER JOB cards in the reader and press START on the reader. To cancel a previous IGCON job press "PSW Restart" on the computer control panel. The IGFILER job execution cards are as follows:

```
//IGFILER        JOB
//SYS004 ACCESS  IGFILER,182 = 'NEWIG'
//               EXEC IGFILER
 /*
 /&
```

The complete IGCON/IGFILER program package is stored on disk during execution. When the user is finished using these programs they must be deleted from the disk. A delete job must be executed to do this. The delete cards are

included with the other program cards. If these are not available the delete job is:

```
//DELETE JOB
//SYSAB2 ACCESS   SDSABS, OCO = 'SYSRES'
//DELETE SDSABS(IGPHASE1,IGPHASE2,IGPHASE3,IGFILER ,GRAPHER,SUBS)
//CONDENSE SDSABS
 /*
 /&
```

b. Filer Operation

The keyboard messages and requested responses for this program are clear enough so that reference to the program instruction manual should be rarely needed. A general discussion of the operation of the program is presented here.

Basically the filer performs two functions: (1). Reads data from the file and (2) Changes the file. To read from the file READ (ALL entries in caps) is entered when requested. Four options are available in this mode: (1) PLOT produces a 100 point graph of the spectrum in terms of relative amplitude and wavelength. (2) PNCH produces a 100 point punched card output of the above graph. The format is five data points per card in F8.3 FORTRAN format with the amplitude and wavelength pair composing each data point, in that order. Thus, there will be 20 cards punched each time the PNCH option is selected. (3) The GRAF option produces a graph of the averaged interferogram data which was Fourier transformed to produce the spectrum plot. (4) The PRNT option prints the numerical values of the magnitude and phase angle points produced by the Fourier transform program. All printed output is produced on the 1403 line printer. To enable examination of the contents of the file a "TABLE OF CONTENTS" option is included which prints out on the line printer the ID number and source description of all the data in the file. When the program is initially executed the a table of

contents of the existing file is automatically produced. Any data in the file can be obtained by typing in the ID number associated with the desired interferometer data. There are two ID request options, "single" and "multiple." The multiple option assumes contiguous IG data output is desired from the first ID input to and including the last ID number entered.

The "EDIT" option supplies three possible functions: "ADD," "REPLACE" and "DELETE". The add and replace functions require a tape containing new IG data to be added or replaced on unit 182 and a new IG file tape on unit 180 as well as the old file on 181. The delete function requires only the old file on 181 and the new file on 180. The add function copies data from the old file and new IG tape and writes it on the new file in proper order. The ordering of IG data is by year, month, day, instrument type, and run serial number for that day. The replace function deletes IG data from the old file and replaces it with new data from the new IG tape having the same ID numbers. The delete function removes IG's from the file.

It is important that a proper file protection procedure be followed when operating on the file tapes to prevent destruction of the data. The ring must always be out of the existing file before it is mounted on 181. When the new file has been formed on unit 180 the ring should be removed immediately and both the old and new files should be saved. Also accurate labeling of the files must be carried out. It is recommended that the same two tapes always be used for the files and IGFILE should be written on the current file label and erased from the old file label. This will prevent confusion in switching old file to new file and new file to old file labels.

6. Obtaining CALCOMP Plots.

A program exists as part of the system which generates a CALCOMP plot of data on punched cards included with the deck. Variability in the X and Y axis

scales is caused by the method of operation of the CALCOMP manufacturer's program which controls the device. Thus experimentation and patience is required of users in setting axis lengths which will be suitable for the range of data values in a spectrum plot. This part of the IG system has not been fully developed since demand for the plots is low. A source listing for the program is included in section V-3. Users desiring specific plot dimensions must manipulate the program to suit their needs.

The CALCOMP program included in the present system produces a 7 inch Y axis and a 14 inch X axis. The run number is drawn at the top of the graph. The data format is 5 points per card in F8.3 format. The deck setup format is given in the program listing. System control cards are supplied with the deck for use with the IBM 7094 IBSYS computer system.

When a deck is set up as desired the package must be delivered to the input drawers in the sub-basement of the Purdue Computer Science Center building. Output plots are returned at the output window in the basement of the CSC in about 24 to 48 hours. CALCOMP users should read the pertinent sections of the Purdue CSC users manual before using the CALCOMP program.

V. Program Documentation

There are three program packages in the IG system. The names of the programs and subroutines are as follows:

1. INTERFEROGRAM CONVERSION PROGRAM(IGCON)

   a. Phase 1: (IGPHASE1)

      Main Program: IGPART1

      Subroutines: READBN
      UNPKID
      UNPKOS
      LOGOUT
      PLOT12H

   b. Phase 2: (IGPHASE2)
      Main Program: IGPART2
      Subroutines: TRANSF
      MAX
      FORT

   c. Phase 3: (IGPHASE3)
      Main Program: IGPART3
      Subroutines: GRAPH1
      NOR
      NND
      MOMPL

2. INTERFEROGRAM FILE CONTROL PROGRAM (IGFILER)
      Main Program: IGFILER
      Subroutines: COPY
      SORT
      SKIP
      RUNBN
      GRAPH1
      NOR
      NND
      MOMPL

3. CALCOMP Deck

Many other subroutines which are supplied by the computer system are used by these programs and are not discussed here. All of the programs listed above must be supplied by the user from tape or cards except for the following: READBN, RUNBN GRAPH1, NOR, NND, MOMPL. These are stored in the users FORTRAN library on the disk. All the remaining programs are listed below. Flow diagrams are included

only for certain complex algorithms whose operation is non-obvious. Listings for programs not included here can be obtained from the LARS library.

```
N IV    MODEL 44  PS            VERSION 2         DATE  68027

      C IGPART1
      C*****************************************************************
      C*        LABORATORY FOR AGRICULTURAL REMOTE SENSING              *
      C*        INTERFEROGRAM DATA PROCESSING PROGRAM                   *
      C*        IG RETRIEVAL AND AVERAGING PHASE                        *
      C*        PHASE 1 OF A THREE PHASE PROGRAM FOR PS 44              *
      C*****************************************************************
            COMMON FDATA,CONTRL,DID,NGO,NP4C,SW2,NR,NIG,NFIRST,LAST,START,BAND
           2,CAL1,CAL2,NDPNA,EOFF,NUM,NTOL,ID,SKIP,NOFST,SEEK,TAPN,CF,NSAM,
           3NDPN,NDPE,NSCAN,NDELTA,ISAM,NCLK,IPAR,NHAVE,NP,ISFRST,NLAST,NPR,
           4 CONT,ISTA,IFINI,NCHAN,NPNCH,INST,NC,SWEEPT,SWEEPL,P4SF,ISTART,
           5 MINSAM,NOFSET
            INTEGER INST,SOURCE(20),IFLAG(5),DID(50),TAPN,RUNUM,A(10),SKPP,
           2TAKE,NDPNA(4),IGBUF(1024),YES,NOFST(4,2),SEEK,SORTNM,RES,CON
            INTEGER IDATA( 600),JDATA(3,600),COUNT,ID(12),IGSUM(1024),
           2 ISAM(10),CONTRL(29),CONT(13),CF,START,SKIP,BLANKS,DUMDUM(2824)
            REAL FDATA(2824),BAND(2),FDID(10),CAL1(4,2),CAL2(4,4,2)
            EQUIVALENCE(IDATA(1),JDATA(1,500))
            EQUIVALENCE(DUMDUM(1),JDATA(1,1),FDATA(1))
            EQUIVALENCE(FDATA(1801),IGSUM(1))
            EQUIVALENCE(FDID(1),DID(16))
            DATA YES,NO,BLANKS,RES,CON/'YES ','NO  ','    ','RES ','CON '/
      10    IF(NGO.LE.0) NGO=1
            IF(NGO.GT.5) NGO=1
            GO TO(20,70,30,280,608),NGO
      C******* INITIAL PROGRAM LOADING START POINT IS 20 ******
      C*
      C******* INSTRUMENT CALIBRATION DATA TABLE ******************************
      C*        CAL1 ARE SWEEP TIME VALUES.  SUBSCRIPTS(T,I) REPRESENT        *
      C*        TIME SETTING T FOR INSTRUMENT TYPE I.                         *
      C*        CAL2 ARE SWEEP LENGTH VALUES WITH SUBSCRIPTS(T,L,I)           *
      C*        REPRESENTING TIME SETTING T, LENGTH SETTING L, FOR            *
      C*        INSTRUMENT I.                                                 *
      C*        INSTRUMENT CODES ARE,  I=1 FOR 195T, I=2 FOR 195E, I=3 FOR P4.*
      C******* 195T VALUES ***************************************************
      20    CAL1(1,1)=.126
            CAL1(2,1)=.254
            CAL1(3,1)=.51
            CAL1(4,1)=1.1
            CAL2(1,1,1)=66.4
            CAL2(1,2,1)=133.
            CAL2(1,3,1)=266.
            CAL2(1,4,1)=500.
            CAL2(2,1,1)=66.6
            CAL2(2,2,1)=130.
            CAL2(2,3,1)=262.
            CAL2(2,4,1)=539.
            CAL2(3,1,1)=65.3
            CAL2(3,2,1)=133.
            CAL2(3,3,1)=263.
            CAL2(3,4,1)=542.
            CAL2(4,1,1)=65.2
            CAL2(4,2,1)=131.
            CAL2(4,3,1)=272.
            CAL2(4,4,1)=542.
      C******* 195E VALUES ***************************************************
            CAL1(1,2)=.0625
            CAL1(2,2)=.127
            CAL1(3,2)=.255
            CAL1(4,2)=.51
            CAL2(1,1,2)=60.6
            CAL2(1,2,2)=120.
            CAL2(1,3,2)=239.
            CAL2(1,4,2)=493.
            CAL2(2,1,2)=61.6
            CAL2(2,2,2 )=122.
            CAL2(2,3,2)=244.
            CAL2(2,4,2)=484.
            CAL2(3,1,2)=57.8
            CAL2(3,2,2)=121.
            CAL2(3,3,2)=244.
            CAL2(3,4,2)=498.
            CAL2(4,1,2)=61.4
            CAL2(4,2,2)=118.
            CAL2(4,3,2)=244.
            CAL2(4,4,2)=500.
      C******* NUMBER OF DATA POINTS EXPECTED (NOMINAL) FOR 195T ************
```

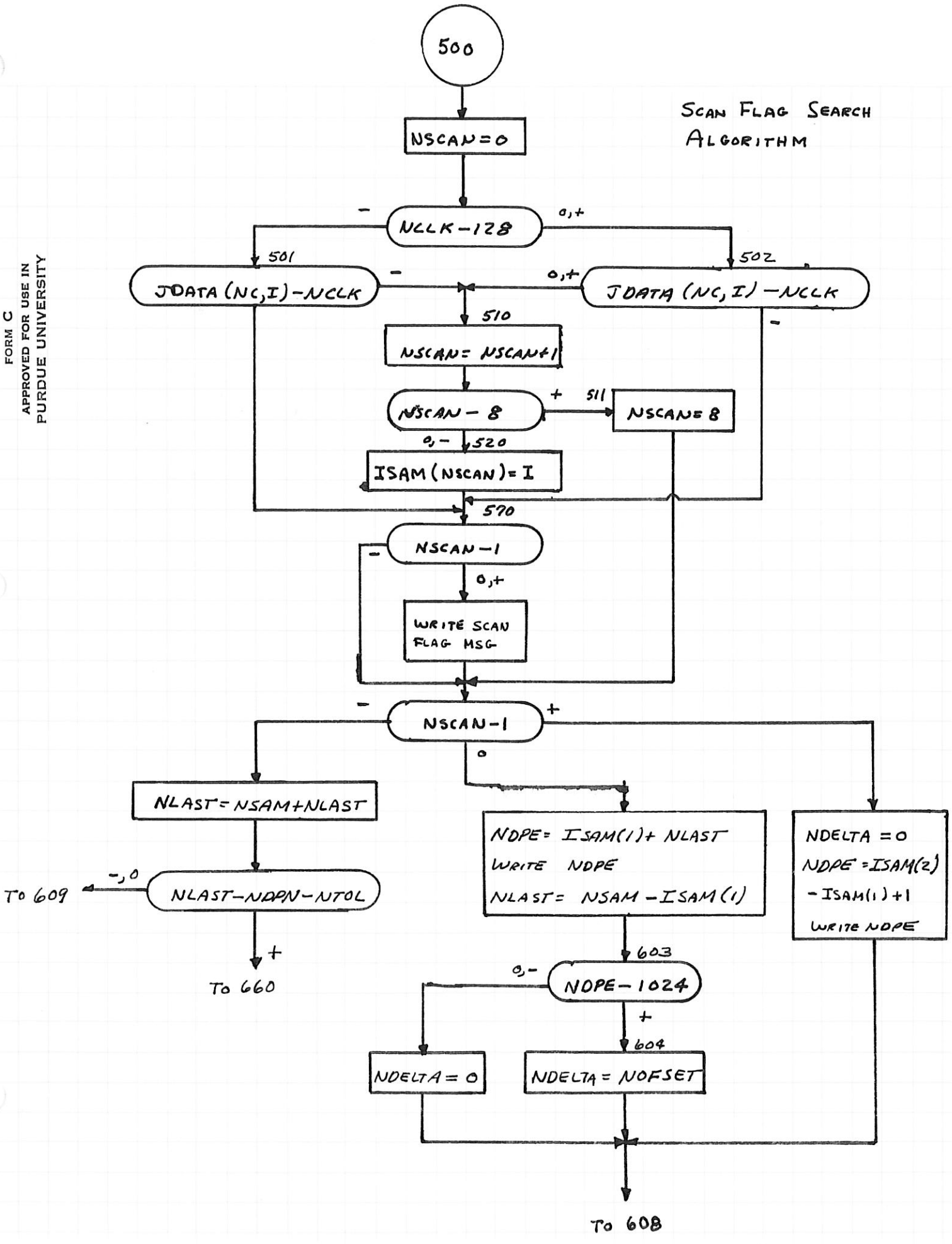
```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

               NDPNA(1)=250
               NDPNA(2)=500
               NDPNA(3)=1000
               NDPNA(4)=2050
      C******* NUMBER OF DATA POINTS FOR OFFSET FROM START OF IG ************
               NOFST(4,1)=512
               NOFST(4,2)=1538
               NOFST(3,2)=512
               NTCL=50
               ISTART=1
      C*          END OF CALIBRATION TABLE
      C*
      C******* CYCLIC AND INITIAL PROGRAM LOADING INITIALIZATION ************
      C*
               GO TO 40
      30       CALL RUNBN(13)
      40       SW2=1
               DO 50 I=1,12
      50       ID(I)=9
               ECFF=0
               WRITE(15,10750)
               NUM=0
                 PAUSE 1
      60       REWIND 13
               ISERST=1
      70       NIG=0
               CF=0
               LPAR=0
               NPR=0
               NP=1
               SKIP=0
               MINSAM=1024
               NLAST=0
               DO 80 I=1,13
      80       CONT(I)=0
               CONT(1)=1
               CONT(13)=1
               DO 90 I=1,1024
      90       IGSUM(I)=0
      C*
      C******* CONTROL CARD READ AND STANDARD OPTION INITIALIZATION SECTION **
      C*
      100      READ(5,10060) CONTRL
               SEEK=0
               IF(CONTRL(29)-1)130,110,120
      110      SKIP=1
               GO TO 300
      120      SEEK=1
      130      READ(5,10650) SOURCE
               IF(CONTRL(13))140,140,149
      140      CONTRL(13)=2
      149      START=CONTRL(13)
               NFIRST=START
               IF(CONTRL(14))150,150,151
      150      CONTRL(14)=9999
      151      IF(SW2)154,154,152
      152      SW2=0
               IF(CONTRL(26))153,153,154
      153      WRITE(15,11000)
                 PAUSE 2
               REWIND 12
      154      ISTA=1
               IF(CONTRL(17).EQ.0) GO TO 155
               ISTA=CONTRL(17)
      155      IFINI=1024
               IF(CONTRL(18).EQ.0) GO TO 156
               IFINI=CONTRL(18)
      156      N=CONTRL(1)
               GO TO(160,161,170),N
      160      NCHAN=2
               NPNCH=1
               BAND(1)=2.0
               BAND(2)=16.0
               INST=-235276829
               NC=2
               GO TO 162
      161      NCHAN=2
```

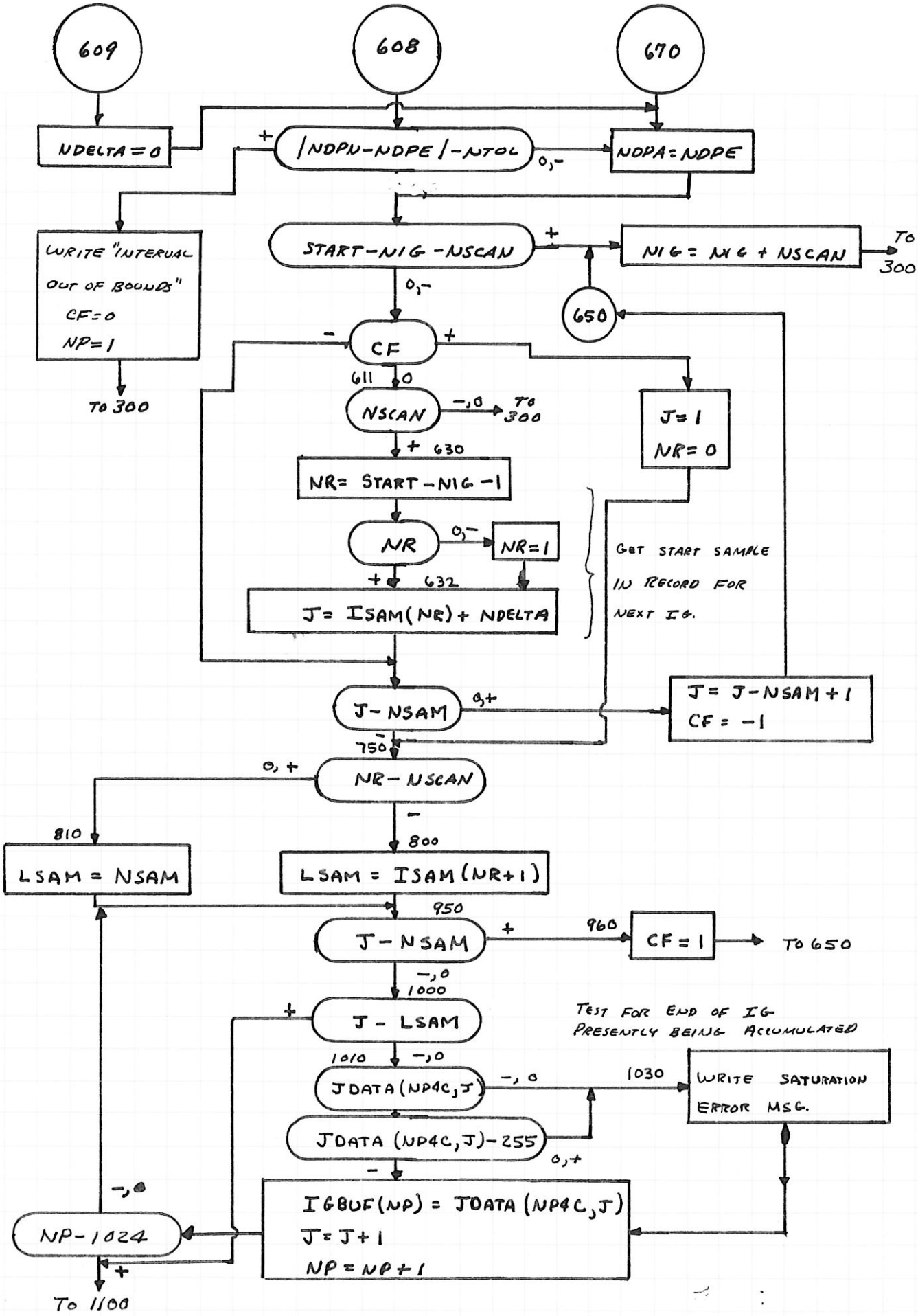
```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

              NPNCH=1
              NC=2
              BAND(1)=2.0
              BAND(2)=6.5
              INST=-235276859
        162   M=CONTRL(8)
              SWEEPT=CAL1(M,N)
              M1=CONTRL(9)
              SWEEPL=CAL2(M,M1,N)
              NDPN=NDPNA(M)*N
              NOFSET=NOFST(M1,N)
              GO TO 171
        170   NCHAN=4
              NPNCH=2
              BAND(1)=.35
              BAND(2)=1.0
              NC=3
              NDPN=948
              INST=1087894592
              P4SF=50.
        171   IF(CONTRL(20))181,181,180
        180   LAST=START
              GO TO 200
        181   LAST=CONTRL(14)
        200   IF(CONTRL(15))210,210,220
        210   CONTRL(15)=1
        220   IF(CONTRL(16))230,230,240
        230   NCLK=50
              GO TO 250
        240   NCLK=CONTRL(16)
        250   CONTINUE
              NP4C=1
        C*
        C******* BULK TAPE BACKUP CONTROL ***************************************
        C*
              WRITE(6,10000)
        280   DO 291 I=1,12
              IF(CONTRL(I)-ID(I))295,291,295
        291   CONTINUE
        292   IF(NUM+1)294,294,293
        293   CALL NOTE(13,IREC)
              IREC=IREC-1
              CALL POINT(13,IREC)
              NUM=NUM-1
              GO TO 292
        294   NUM=0
              GO TO 300
        295   IF(NUM-1)300,397,397
        C*
        C******* TAPE READ AND ERROR RESPONSE SECTION ****************************
        C*
        300   COUNT=300
              IBYTES=4*COUNT
              IF(ISTART.EQ.1) GO TO 305
              CALL LOGOUT
        305   CALL READBN(13,IDATA,IBYTES,IERR)
              IF(ISTART.EQ.1) GO TO 307
              CALL LOGIN
        307   ISTART=0
              COUNT=IBYTES/4
              IF(IERR)350,350,310
        310   GO TO(314,311,312,313,313),IERR
        311   NUM1=NUM+1
              WRITE(6,12100) NUM1
              IPAR=IPAR+1
              GO TO 350
        312   NUM1=NUM +1
              WRITE(6,12150)    NUM1
              GO TO 350
        313   WRITE(6,12200) IERR
              STOP
        C*
        C******* END OF FILE MARK RESPONSE SECTION ******************************
        C*
        314   WRITE(6,10050)
              IF(ECFF)320,320,315
        315   WRITE(15,11600)
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

              IF(CONTRL(28).NE.0) GO TO 317
              WRITE(6,13500)
              WRITE(15,13500)
        316   IN=BLANKS
              READ(15,12600) IN
              IF(IN.EQ.YES) GO TO 317
              IF(IN.EQ.NO) GO TO  300
              WRITE(15,13400)
              GO TO 316
        317   IF(CONTRL(26).EQ.2) GO TO 30
              DO 318 I=1,49
        318   DID(I)=0
              DID(50)=666666
              WRITE(12) DID
              WRITE(6,13300)
              WRITE(15,13300)
              CALL RUNBN(12)
              GO TO 30
        320   EOFF=1
              NUM=0
              IF(SKIP)330,330,332
        330   NHAVE=NIG-1
              IF(NIG)332,332,331
        331   WRITE(6,12300) NHAVE
              GO TO 1300
        332   SKIP=0
              IF(CONTRL(29).EQ.1) GO TO 70
              GO TO 300
        340   IF(SKIP)70,70,300
        350   EOFF=0
              IF(COUNT-3)360,360,400
        360   IF(COUNT)361,361,370
        361   WRITE(6,10030) COUNT
              WRITE(15,10030) COUNT
              WRITE(15,12700)
              PAUSE 6
              CF=0
              NP=1
              GO TO 300
        C*
        C******* DATA UNPACK SECTION ****************************************
        C*
        370   CALL UNPKID(IDATA,ID)
              WRITE(6,10030) COUNT
              WRITE(6,10100) ID
              NUM=0
              ISFRST=0
              DO 380 I=1,12
              IF(CONTRL(I)-ID(I))381,380,381
        380   CONTINUE
              GO TO 300
        381   IF(SEEK)382,382,397
        382   WRITE(6,12400)(CONTRL(I), I=1,12)
              WRITE(15,12500)ID,(CONTRL(I), I=1,12)
        383   IN=BLANKS
              READ(15,12600) IN
              IF(IN.EQ.YES) GO TO 386
              IF(IN.EQ.NO) GO TO 384
              WRITE(15,12800)
              GO TO 383
        384   WRITE(15,13400)
        385   IN=BLANKS
              READ(15,12600) IN
              IF(IN.EQ.RES) GO TO 60
              IF(IN.EQ.CON) GO TO 397
              WRITE(15,12800)
              GO TO 385
        386   WRITE(15,12900)
        390   IN=BLANKS
              READ(15,12600) IN
              IF(IN.EQ.YES) GO TO 395
              IF(IN.EQ.NO) GO TO 300
              WRITE(15,12800)
              GO TO 390
        395   WRITE(15,13000)
              DO 396 I=1,12
              READ(15,13100) CONTRL(I)
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      396    CONTINUE
             GO TO 300
      397    SKIP=1
             GO TO 300
      400    IF(ISFRST)403,403,401
      401    WRITE(15,13200)
             WRITE(6,13200)
             ISFRST=0
             DO 402 I=1,12
      402    ID(I)=0
             GO TO 382
      403    IF(EOFF)404,404,401
      404    NUM=NUM+1
             IF(SKIP)410,410,300
      410    CONTINUE
             CALL UNPK08(IDATA,JDATA,NCHAN,3,COUNT)
             IF(CONTRL(25))470,470,420
      C******* SCAN FLAG INSERTION SECTION ******
      420    READ(5,10800)NUMREC,NFLG,(IFLAG(I) , I=1,NFLG)
             IF(NUM-NUMREC)430,440,430
      430    PAUSE 99
             GO TO 420
      440    IF(NFLG)470,470,444
      444    DO 450 I=1,NFLG
             NFIX= IFLAG(I)
      450    JDATA(NC,NFIX) = 200
      470    NSAM=(COUNT*4)/NCHAN
      C*
      C******* SCAN START FLAG SEARCH SECTION **********************************
      C*
             NSCAN=0
      500    DO 570 I=1,NSAM
             IF(NCLK-128)501,502,502
      C******* TEST FOR DATA SATURATION ******
      501    IF(JDATA(NC,I)-NCLK) 510,510,570
      502    IF(JDATA(NC,I)-NCLK)570,510,510
      510    NSCAN=NSCAN+1
             IF(NSCAN-8)520,520,511
      511    NSCAN=8
             GO TO 591
      520    ISAM(NSCAN)=I
      570    CONTINUE
             IF(NSCAN-1)580,590,591
      580    CONTINUE
             GO TO 600
      590    WRITE(6,10400)NUM,COUNT,NSCAN,(ISAM(I), I=1,NSCAN)
             GO TO 600
      591    WRITE(6,10410)NUM,COUNT,NSCAN,(ISAM(I), I=1,NSCAN)
      600    IF(NSCAN-1)602,601,607
      601    NDPE=ISAM(1) + NLAST
             WRITE(6,10900)NDPE
             NLAST = NSAM - ISAM(1)
             GO TO 603
      602    NLAST=NSAM + NLAST
             IF(NLAST-NDPN-NTOL)609,609,660
      603    IF(NDPE-1024)605,605,604
      604    NDELTA=NOFSET
             GO TO 608
      605    NDELTA=0
             GO TO 608
      607    NDELTA=0
             NDPE=ISAM(2)-ISAM(1)+1
             WRITE(6,10900)NDPE
      608    CONTINUE
      C******* TEST FOR ACCEPTABLE NUMBER OF POINTS IN IG *******
      680    IF( IABS(NDPN-NDPE)-NTOL)670,670,660
      660    WRITE(6,11700)
             CF=0
             NP=1
             GO TO 300
      609    NDELTA=0
      670    NDPA=NDPE
             IF(START-NIG-NSCAN)610,610,650
      650    NIG=NIG+NSCAN
             GO TO 300
      C******* CF IS CONTINUATION FLAG ******
      C*        CF=-1 = DESIRED IG FOUND BUT START POINT NOT IN
```

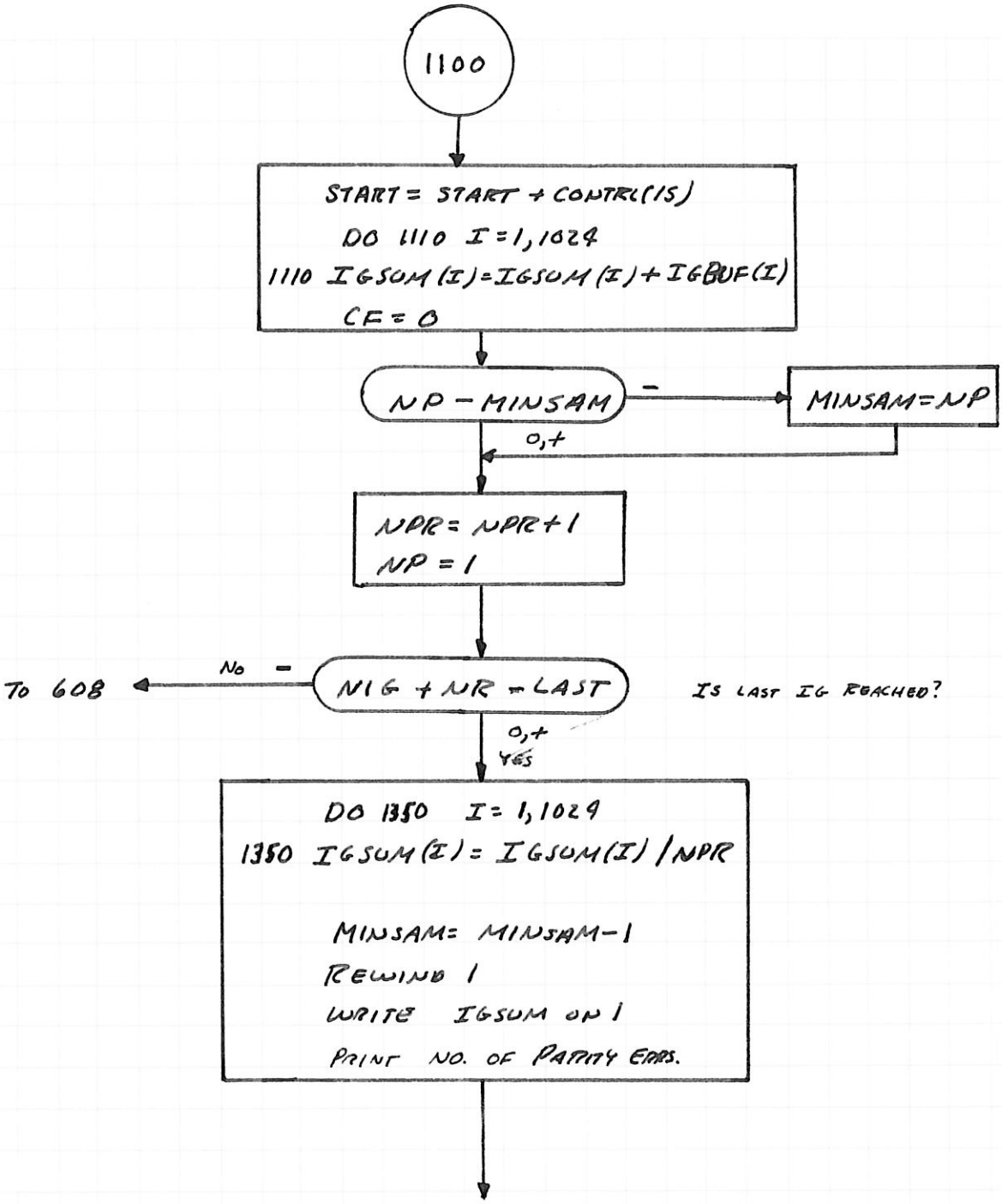
Scan Flag Search Algorithm





ALGORITHM FOR ASSEMBLING DATA FOR ONE INTERFEROGRAM

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027
      C*         THIS TAPE RECORD.
      C*         CF=0 = IG IS NOT BEING ACCUMULATED.
      C*         CF=1 = IG IS BEING ACCUMULATED AND CONTINUES ON NEXT
      C*         TAPE RECORD.
      610    IF(CF)640,611,620
      611    IF(NSCAN)300,300,630
      620    J=1
             NR=0
             GO TO 750
      630    NR=START-NIG
             IF(NR)631,631,632
      631    NR=1
      632    J=ISAM(NR) + NDELTA
      640    IF(J-NSAM)750,700,700
      700    J=J-NSAM+1
             CF=-1
             GO TO 650
      750    IF(NR-NSCAN)800,810,810
      800    LSAM=ISAM(NR+1)-1
             GO TO 950
      810    LSAM=NSAM
      950    IF(J-NSAM)1000,1000,960
      960    CF=1
             GO TO 650
      1000   IF(J-LSAM)1010,1010,1100
      C*
      C******* DATA AVERAGING SECTION ****************************************
      C*
      1010   IF(JDATA(NP4C,J))1030,1030,1020
      1020   IF(JDATA(NP4C,J)-255)1050,1030,1030
      1030   WRITE(6,11800)J,JDATA(NP4C,J)
      1050   IGBUF(NP)=JDATA(NP4C,J)
             J=J+1
             NP=NP+1
             IF(NP-1024)1080,1080,1100
      1080   GO TO 950
      1100   START=START+CONTRL(15)
             DO 1110 I=1,1024
      1110   IGSUM(I)=IGSUM(I)+IGBUF(I)
             CF=0
             IF(NP-MINSAM)1150,1200,1200
      1150   MINSAM=NP
      1200   NPR=NPR+1
             NP=1
             IF(NIG+NR-LAST)608,1300,1300
      1300   DO 1350 I=1,1024
      1350   IGSUM(I)=IGSUM(I)/NPR
             MINSAM=MINSAM-1
             REWIND 1
             WRITE(1) IGSUM
             WRITE(6,13700) IPAR
             IF(CONTRL(22))1370,1370,1360
      C******* DATA PRINT SECTION **********
      1360   WRITE(6,11100)
             DO 1361 I2=1,MINSAM,20
             I3=I2+19
      1361   WRITE(6,11200)I2,(IGSUM(I1), I1=I2,I3)
      C******* DATA GRAPH SECTION **********
      1370   WRITE(6,10350) NPR
             WRITE(6,10370)
             WRITE(6,10390)
             MAX=0
             DO 1372 I=1,MINSAM
             IF(IGSUM(I).LE.MAX) GO TO 1372
             MAX=IGSUM(I)
             MAXPT=I
      1372   CONTINUE
             IF(MAXPT.GT.200) GO TO 1373
             LST=1
             IFIN=200
             GO TO 1377
      1373   IF(MAXPT.LT.824) GO TO 1375
             LST=824
             IFIN=1024
             GO TO 1377
      1375   LST=MAXPT-100
             IFIN=MAXPT+100
```



INTERFEROGRAM AVERAGING SECTION

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      1377  IF(CONTRL(23).EQ.0) GO TO 1380
            IST=ISTA
            IFIN=IFINI
      1380  CALL PLOT12(IGSUM,CONT,IST,IFIN,1)
      1400  IF(CONTRL(24))2000,2000,1420
      C******* DATA PUNCH SECTION *********
      1420  WRITE(15,13600)
            PAUSE 2
            IFIN2=IFIN-20
            DO 1500 IT=IST,IFIN2,20
            IT1=IT+19
      1500  WRITE(7,10221) (IGSUM(II), II=IT,IT1)
            READ(5,10200)
            WRITE(15,13800)
            PAUSE 3
      C*
      C******* INITIAL ID RECORD FORMATION SECTION ****************************
      C*
      2000  RUNUM=CONTRL(10)*100 + CONTRL(11)*10 + CONTRL(12)
            SORTNM=RUNUM + CONTRL(1)*1000+ CONTRL(5)*10000 + CONTRL(4)*100000
            SORTNM=SORTNM + CONTRL(3)*10**6 + CONTRL(2)*10**7 +CONTRL(7)*10**8
            DO 2540 I=3,9
            II=10-I
      2540  RUNUM = RUNUM + CONTRL(II)*10**I
            DID(1)=SORTNM
            DID(2)=RUNUM
            DID(3)= 10*CONTRL(2) + CONTRL(3)
            DID(4) = 10*CONTRL(4) + CONTRL(5)
            DID(5) = 10*CONTRL(6) + CONTRL(7)
            DID(8)= CONTRL(1)
            DID(9)= NP4C
            DID(10)= NPR
            DID(11)= NFIRST
            DID(13)= NDPA
            DID(14)= C
            DID(15)= 0
            FDID(1)= SWEEPT
            FDID(2)= SWEEPL
            FDID(3)= BAND(1)
            FDID(4)= BAND(2)
            FDID(8)=0.
            DID(12)=MINSAM
            FDID(9)=C.
            FDID(10)=0.
            DID(26)= BLANKS
            DID(27)= BLANKS
            DID(28)= INST
            DO 2550 I=1,20
      2550  DID(I+28)= SOURCE(I)
            DID(49)=0
      C*
      C******* END OF IGCON PART 1 ********************************************
      C*
      C*       COMPLETE OVERLAY BY PART 2 INITIATED HERE
            CALL LINK('IGPHASE2')
      C*
      C******* FORMAT STATEMENT SECTION ****************************************
      C*
      10000 FORMAT('1            LABORATORY FOR AGRICULTURAL REMOTE SENSING '
           2 // '            DIGITAL DATA SYSTEM ' // '                INTERFEROME
           2TER DATA PROCESSING PROGRAM')
      10030 FORMAT('0            COUNT=',I5)
      10050 FORMAT('0            END OF FILE MARK ENCOUNTERED  '/)
      10060 FORMAT(12I1,7I4,10(3X,I1))
      10100 FORMAT('0            ID RECORD= ',12(I1,1X)/)
      10200 FORMAT(80X)
      10220 FORMAT(10F8.2)
      10221 FORMAT(2CI4)
      10350 FORMAT('1            GRAPH OF INTERFEROGRAM AVERAGED OVER ',I4,' SC
           2ANS   ')
      10370 FORMAT('CSAMPLE NUMBER ',45X,' DATA VALUE ' )
      10390 FORMAT( 9X,'0',19X,'50',17X,'100',17X,'150',17X,'200',17X,'250')
      10400 FORMAT('              DATA RECORD NO.' ,I4,', BLOCK COUNT= ',I3,', '
          02,I2,' SCAN START FLAG FOUND AT SAMPLE ',I3)
      10410 FORMAT('              DATA RECORD NO.' ,I4,', BLOCK COUNT= ',I3,', '
           2,I2,' SCAN START FLAGS FOUND AT SAMPLES ',8(I3,','))
      10450 FORMAT('              DATA RECORD NO.' ,I4,', BLOCK COUNT= ',I3,', '
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      2,'NC SCAN START FLAGS FOUND IN THIS RECORD'/)
10650 FORMAT(20A4)
10750 FORMAT('  MOUNT BULK DATA TAPE ON 183 ',/, '  LOAD CONTROL CARDS '
     2,/, '  ENTER EOB')
11000 FORMAT('    MOUNT BLANK IG DATA TAPE ON 182, RING IN.'/'    ENTER
     2EOB.')
10800 FORMAT(10I4)
10900 FORMAT('                NUMBER OF SAMPLES FOUND IN LAST IG= ',I5/)
11100 FORMAT('1   SAMP     NUMERICAL VALUES OF AVERAGED INTERFEROGRAM SAM
     2PLES')
11200 FORMAT(4X,I4,9X,20(I3,2X))
11600 FORMAT(/'  LAST IG ON THIS TAPE')
11700 FORMAT('C               SCAN FLAG INTERVAL OUTSIDE EXPECTED BOUNDS, IG
     2 SKIPPED'/)
11800 FORMAT('       *****DATA POINT NO. ',I3,' = ',I3,' POSSIBLE SATURA
     2TION ERROR'/)
12100 FORMAT( '    PERMANENT PARITY CHECK ERROR CONDITION EXISTS AFTER R
     2EADING RECORD ',I4,                                        /'    T
     2HE DATA SET IS POSITIONED JUST PAST THE BLOCK CONTAINING THE ERROR
     3'/'    PROCESSING CONTINUED'/)
12150 FORMAT( '          READ OPERATION FOR RECORD',I4,' WAS TERMINATED WI
     3THOUT TRANSMITTING ALL THE DATA.'/'    THE POSITION OF THE DATA SE
     3T IS NOT KNOWN.'/'    PROCESSING CONTINUED'/)
12200 FORMAT(/'    ERROR ',I4,/'    IMPOSSIBLE.'/'    SEE PG. 66 OF IBM
     2C28-6812-1.'/)
12300 FORMAT('C               NUMBER OF INTERFEROGRAMS IN THIS RUN= ',I3)
12400 FORMAT('                ID NUMBERS DO NOT AGREE',/,'              CARD I
     2D = ',12(I1,1X)/)
12500 FORMAT(/'  ID NUMBERS DO NOT AGREE',/,'  TAPE ID = ',12(I1,1X),/,'
     2  CARD ID = ',12(I1,1X),/,'  DO YOU WISH TO ',/,'  PROCESS THIS RU
     3N ',/,'  ENTER ''YES'' OR ''NO''   ')
12600 FORMAT(A3)
12700 FORMAT(/'    SUGGEST CLEANING TAPE DRIVE'/'    ENTER EOB TO CONTIN
     2UE')
12800 FORMAT(/'  INVALID RESPONSE'/'  RETYPE')
12900 FORMAT(/'  RE-READ ID NUMBER '/'  ENTER ''YES'' OR ''NO''  ')
13000 FORMAT(/'  ENTER CORRECT ID ( 12 DIGITS'/'  WITH EOB AFTER EACH DI
     2GIT) ')
13100 FORMAT(I1)
13200 FORMAT(/'    NO ID RECORD FOR THIS RUN')
13300 FORMAT(/'    END OF DATA RECORD WRITTEN '/'    ON OUTPUT TAPE')
13400 FORMAT(/'    RESTART JOB OR CONTINUE '/'    ENTER ''REST'' OR ''CO
     2NT''    ')
13500 FORMAT(/'    2 EOF MARKS FOUND AFTER RUN NOT FLAGGED AS LAST ON CO
     2NTROL CARD.'/'    IS THIS THE END OF JOB '/'    ENTER ''YES'' OR '
     3'NO''    ')
13600 FORMAT(/'    AVERAGED IG DATA PUNCH CALLED FOR.'/'    LOAD BLANK C
     2ARDS, ENTER EOB.')
13700 FORMAT('0',11X,I3,' BULK TAPE PARITY CHECK ERRORS ENCOUNTERED IN T
     2HIS RUN.')
13800 FORMAT( '    RELCAD CONTRL CDS.')
13900  FORMAT('0')
       END
```

N IV MODEL 44 PS VERSION 2 DATE 68027

COMMON BLOCK / / MAP SIZE C02F5C

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CCCCC0	CUMCUM	CCCCC0	JDATA	CCCCC0	IDATA	001764
002C20	DID	CC2C94	FDID	002CDC	NGO	002D5C
002D64	NR	C02D68	NIG	C02D6C	NFIRST	002D70
C02D78	BAND	002D7C	CAL1	002D84	CAL2	002DA4
C02E34	NUM	CC2E38	NTOL	002E3C	ID	002E40
002E74	SEEK	C02E94	TAPN	002E98	CF	002E9C
002EA4	NDPE	C02EA8	NSCAN	CC2EAC	NDELTA	002EB0
CC2EDC	IPAR	002EE0	NHAVE	002EE4	NP	002EE8
002EFC	NPR	CC2EF4	CCNT	002EF8	ISTA	002F2C
002F34	NPNCH	CC2F38	INST	002F3C	NC	002F40
C02F48	P4SF	C02F4C	ISTART	C02F50	MINSAM	002F54

SCALAR MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CC0444	NC	CC0448	BLANKS	00044C	RES	000450
000458	N	CC045C	M	000460	M1	000464
00046C	IBYTES	CC0470	IERR	000474	NUM1	000478
CC0480	NFLG	000484	NFIX	CC0488	NDPA	00048C
CC0494	I2	CC0498	I3	00049C	I1	0004A0
C004A8	IST	CC04AC	IFIN	0004B0	IFIN2	0004B4
C004BC	II	CC04C0	RUNUM	0004C4	SORTNM	0004C8

ARRAY MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
0004CC	IFLAG	CC051C	A	000530	IGBUF	000558

SUBPROGRAMS CALLED

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
C01558	IBCOM=	00155C	NOTE	001560	POINT	001564
C0156C	LOGIN	001570	UNPKID	001574	UNPK08	001578
C01580	FIXPI=	001584				

LABEL MAP

LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
00161A	20	CC1694	30	00182A	40	001838
0018B6	70	0C18D8	80	001928	90	00196A
CC19DE	120	0019F4	130	CC1A04	140	001A38
CC1A68	151	001A78	152	CC1A88	153	001AA4
C01AFE	156	001B24	160	C01B6C	161	001B9E
001C5C	171	001CA4	180	001CB4	181	001CC6
C01CE2	220	CC1CF2	230	001DC2	240	001D18
001D50	291	CC1D80	292	001D96	293	001DAE
001E02	300	CC1E1A	305	001E4E	307	001E78
C01EDC	312	001F26	313	001F62	314	001F92
002014	317	CC2072	318	002C90	320	00211C
C02160	332	00218E	340	CC21B2	350	0021C4
C021FC	370	C0226C	380	002300	381	00231C
C023DC	384	C0243A	385	CC2454	386	0024B2
CC252A	396	CC2584	397	00259C	400	0025B2
C02608	403	002638	404	002648	410	002668
C026FC	440	CC2714	444	CC2720	450	00273C
C0279E	501	CC27BE	502	0027F6	510	002828
C02862	570	002876	580	CC28A6	590	0028AC
CC2990	601	0029AE	602	0029F2	603	002A1C
C02A46	607	C02A5C	608	002A9C	680	002A9C
002AEE	670	C02AFE	650	CC2B1E	610	002B34
002B6C	630	002B7E	631	CC2B9A	632	002BAA
002BD6	750	CC2BFC	800	002C10	810	002C32
C02C52	1CCC	002C68	1010	CC2C78	1020	002CA6
002D30	1080	CC2D8A	1100	002D9C	1110	002DAC
CC2EC2	1300	C02E32	1350	C02E3E	1360	002ED8
002F76	1372	CC3002	1373	003C40	1375	003068
0030AA	1400	CC30B8	1420	003CC8	1500	00311A
003260	255C	003376	10000	CC33B6	10030	003454
C034A8	101CC	CC34C2	10200	C034EC	10220	0034F6
003512	10370	CC355C	10390	003586	10400	0035B8
CC368C	10650	C036F6	10750	003704	11000	003752
0037AA	11100	CC37E8	11200	003832	11600	003848
0038B8	12100	003906	12150	C039CC	12200	003A6E
003AFA	12500	CC3B48	12600	003BDE	12700	003BE8
C03C58	13CC0	003C92	13100	003CDC	13200	003CE6
C03D52	135CC	003D9C	13600	003E2A	13700	003E80
003EEC						

```
AN IV     MODEL 44  PS              VERSION 2          DATE  68031

      C*****************************************************************
      C*        JACQUES FOURIER TRANSFORM SECTION                      *
      C*        PHASE 2 OF A THREE PHASE PROGRAM FOR PS 44             *
      C*****************************************************************
            COMMON FDATA,CONTRL,DID,NGO,NP4C,SW2,NR,NIG,NFIRST,LAST,START,BAND
           2,CAL1,CAL2,NDPNA,EOFF,NUM,NTOL,ID,SKIP,NOFST,SEEK,TAPN,CF,NSAM,
           3NDPN,NDPE,NSCAN,NDELTA,ISAM,NCLK,IPAR,NHAVE,NP,ISFRST,NLAST,NPR,
           4 CONT,ISTA,IFINI,NCHAN,NPNCH,INST,NC,SWEEPT,SWEEPL,P4SF,ISTART,
           5 MINSAM,NOFSET
            REAL FDATA(2824),BAND(2),FDID(10),CAL1(4,2),CAL2(4,4,2)
            INTEGER CONTRL(29),DID(50),IDATA(2824),NDPNA(4),ISAM(10),CONT(13)
            INTEGER ID(12),NOFST(4,2)
            EQUIVALENCE(FDATA(1),IDATA(1))
            EQUIVALENCE(DID(16),FDID(1))
            MINSAM=DID(12)
      205   DO 210 I=1,MINSAM
            FDATA(2*I-1)=IDATA(I+1800)
      210   FDATA(2*I)=0.
      C*        IHALF = 0 RESULTS IN TRANSFORM OF 1024 PTS.            *
      C*        IHALF = 1 RESULTS IN TRANS. OF CENTER 512 PTS IF MAX.  *
      C*        VALUE FALLS IN CENTER HALF.  IF NOT TAKES 1024 PT. TRANS.  *
            IHALF=CONTRL(21)
            IGRAPH=0
            IPRINT=0
            WRITE(6,1010)
            CALL TRANSF(FDATA,FMSD,FMST,MINSAM,MAXPNT,NPTS,IHALF,IGRAPH,IPRINT
           2)
            FDID(5)=FMSD
            DID(7)=MAXPNT
            FDID(6)=FMST
            DID(6)=NPTS
            WRITE(6,1015)
            WRITE(6,1020)
      C*        COMPLETE OVERLAY BY PART 3 INITIATED HERE.             *
            CALL LINK('IGPHASE3')
       1010 FORMAT('             FOURIER TRANSFORM PROG. ENTERED.')
      1015  FORMAT('0            PROG. CHK. ERRORS DUE TO UNDERFLOW CONDITION C
           2AUSED BY SMALL VALUES OF DATA.'/'                THIS CONDITION IS ACC
           3EPTABLE.')
      1020  FORMAT('             TRANSFORM COMPLT.')
            END
```

```
AN IV    MODEL 44  PS              VERSION 2          DATE  68031

                                COMMON BLOCK /       / MAP SIZE  002F5C
    LOCATION        SYMBOL      LOCATION        SYMBOL      LOCATION        SYMBOL      LOCATION
    000000          IDATA       000000          CONTRL      002C20          DID         002C94
    002D5C          NP4C        002D60          SW2         002D64          NR          002D68
    002D70          LAST        002D74          START       002D78          BAND        002D7C
    002DA4          NDPNA       002E24          EOFF        002E34          NUM         002E38
    002E40          SKIP        002E70          NOFST       002E74          SEEK        002E94
    002E9C          NSAM        002EA0          NDPN        002EA4          NDPE        002EA8
    002EB0          ISAM        002EB4          NCLK        002EDC          IPAR        002EE0
    002EE8          ISFRST      002EEC          NLAST       002EF0          NPR         002EF4
    002F2C          IFINI       002F30          NCHAN       002F34          NPNCH       002F38
    002F40          SWEEPT      002F44          SWEEPL      002F48          P4SF        002F4C
    002F54          NOFSET      002F58

                                SCALAR MAP
    LOCATION        SYMBOL      LOCATION        SYMBOL      LOCATION        SYMBOL      LOCATION
    0000E8          IHALF       0000EC          IGRAPH      0000F0          IPRINT      0000F4
    0000FC          MAXPNT      000100          NPTS        000104

                                SUBPROGRAMS CALLED
    LOCATION        SYMBOL      LOCATION        SYMBOL      LOCATION        SYMBOL      LOCATION
    000108          TRANSF      00010C          LINK        000110

                                LABEL MAP
    LOCATION        LABEL       LOCATION        LABEL       LOCATION        LABEL       LOCATION
    000162            210       0001AE            1010      00025E            1015      000294

L MEMORY REQUIREMENTS 000394 BYTES

R HIGHEST SEVERITY CODE WAS  0
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      C IGPART3
      C*****************************************************************************
      C*        LABORATORY FOR AGRICULTURAL REMOTE SENSING                         *
      C*        INTERFEROGRAM DATA PROCESSING SYSTEM                               *
      C*        CALIBRATION AND REFORMATTING SECTION                               *
      C*        PHASE 3 OF A 3 PHASE PROGRAM FOR PS 44                             *
      C*****************************************************************************
            COMMON FDATA,CCNTRL,DID,NGO,NP4C,SW2,NR,NIG,NFIRST,LAST,START,BAND
           2,CAL1,CAL2,NDPNA,EOFF,NUM,NTOL,ID,SKPP,NOFST,SEEK,TAPN,CF,NSAM,
           3NDPN,NDPE,NSCAN,NDELTA,ISAM,NCLK,IPAR,NHAVE,NP,ISFRST,NLAST,NPR,
           4 CCNT,ISTA,IFINI,NCHAN,NPNCH,INST,NC,SWEEPT,SWEEPL,P4SF,ISTART,
           5 MINSAM,NOFSET
            REAL Y(512),X(512),FDATA(2824),FDID(10),CAL1(4,2),CAL2(4,4,2)
            INTEGER BAND(2),IGSUM(1024),NDPNA(4),ISAM(10),CONT(13),ID(12)
            INTEGER CONTRL(29),DID(50),RUNUM,SKIP,TAPN,BLANKS,TAKE,NOFST(4,2)
            REAL VAL1(3),VAL2(3),LAMUP,LAMLO
            EQUIVALENCE(FDATA(1801),IGSUM(1)),(DID(16),FDID(1))
            DATA TAKE,SKIP,BLANKS/'TAKE','SKIP','    '/
      C******* INITIALIZATION SECTION ******
      1     RUNUM=DID(2)
            VAL1(1)=22.
            VAL2(1)=2.
            VAL1(2)=7.
            VAL2(2)=2.
            VAL1(3)=2.6
            VAL2(3)=.4
            IEQ=CONTRL(1)
            LAMUP=VAL1(IEQ)
            LAMLO=VAL2(IEQ)
            IF(DID(1).EQ.2) LAMUP=7.
            NPTS=DID(6)
            NNPTS=NPTS*2
            WRITE(1) (FDATA(I), I=1,1024)
            REWIND 1
            SW4=0
            DO 42 I=2,NNPTS
      42    FDATA(I)=FDATA(2*I-1)
      C******* SCALE FACTOR CALCULATION ******
            IF(DID(8)-2)55,55,57
      55    FNUM=DID(13)
            FNPTS=NNPTS
            SCALF=   FDID(2)*FNPTS/FNUM
            GO TO 70
      57    SCALF=50.
      C*****************************************************************************
      C*        RELATIVE FREQUENCY PLOT SECTION                                    *
      C*****************************************************************************
      70    MAXPT=DID(7)
            FDID(8)=SCALF
            L=0
      C******* DATA POINT SELECTION ******
            IF(MAXPT-50)71,72,72
      71    LOWPT=1
            LPTPLS=100
            GO TO 76
      72    IF(MAXPT-NPTS+50)74,73,73
      73    LPTPLS=NPTS
            LOWPT=LPTPLS-99
            GO TO 76
      74    LOWPNT=MAXPNT-50
            LPTPLS=LOWPT+99
      76    DO 80 I =LOWPT,LPTPLS
            L=L+1
            X(L)=I
      80    Y(L)=FDATA(I)
            IF(NPTS-100)85,83,83
      83    NPLOT=100
            GO TO 90
      85    NPLOT=NPTS
      90    WRITE(6,1045)
            CALL GRAPH1(0,X,Y,NPLOT,0.0,0.0,1,0,0)
            WRITE(6,1047)
            IGO=1
            GO TO 340
      C*****************************************************************************
      C*        FREQ. TO WAVELENGTH CONVERSION & PLOT SECTION                      *
      C*****************************************************************************
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027
      100    L=0
             DO 110 I=1,NPTS
             L=L+1
             FI=I
      110    X(L)=SCALF/FI
             DO 150 J=1,NPTS
             IF(X(J)-LAMUP)155,155,150
      150    CONTINUE
      155    NUP=J
             DO 160 K=J,NPTS
             IF(X(J)-LAMLC)165,165,160
      160    CONTINUE
      165    NLC=K
             NPLOT=NLC-NUP+1
             IF(NPLOT-100)168,168,175
      168    LIM=NLC
             GO TO 176
      175    LIM=NUP+99
             NPLOT=100
      176    CONTINUE
             I2=0
             DO 180 I=NUP,LIM
             I2=I2+1
             FI=I
             X(I2)=SCALF/FI
      180    Y(I2)=FDATA(I)
             WRITE(6,1050)
             CALL GRAPH1(C,X ,Y ,NPLOT,0.0,0.0,1,0,0)
             WRITE(6,1051)
             IGO=2
             GO TO 340
      190    WRITE(6,5000) NPLOT,NUP,NLC,SCALF
      C*************************************************************************
      C*        WAVELENGTH INTERPOLATION & PLOT SECTION                        *
      C*************************************************************************
      1661   NGOOD=NLC-NUP+1
             NPLOT=100
      C******* INTERPOLATION ALGORITHM ******
      200    RLAM=LAMUP-LAMLC
             FNPLOT=NPLOT-1
             DLAM=RLAM/FNPLOT
             FLAM=LAMUP
             DO 410 I5=1,NPLOT
             I6=NPLOT-I5+1
             FRE=SCALF/FLAM
             IFRE=FRE
             FRE2=IFRE
             VALUE=FDATA(IFRE)+ (FDATA(IFRE + 1)- FDATA(IFRE))*(FRE-FRE2)
             X(I6)=FLAM
             FLAM=FLAM-DLAM
      410    Y(I6) =VALUE
             IF(SW4)235,235,240
      235    WRITE(6,1050)
             CALL GRAPH1(C,X ,Y ,NPLOT,0.0,0.0,1,0,0)
             WRITE(6,1051)
             IGO=3
             GO TO 340
      430    WRITE(6,2050)
      C******* GRAPH DATA PUNCH SECTION ******
             IF(CONTRL(27))250,250,151
      151    WRITE(15,2025)
               PAUSE 3
      240    WRITE(7,2030)  RUNUM,NPLOT
             DO 243 NN=1,NPLOT,5
             NNN=NN+4
      243    WRITE(7,2040) (X (I),Y (I), I=NN,NNN  )
             WRITE(15,2090)
             PAUSE 4
      250    CONTINUE
             WRITE(15,1150) RUNUM
             IF(CONTRL(26)-1)265,251,300
      C*************************************************************************
      C*        DATA TAPE WRITE SECTION                                        *
      C*************************************************************************
      251    WRITE(15,1130)
      252    READ(15,1160) IN
             IF(IN-TAKE)255,265,255
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      255    IF(IN-SKIP)260,300,260
      260    WRITE(15,1140)
             GO TO 252
      265    DID(14)=NPLOT
             DID(50)=999999
      280    WRITE(12) DID
             WRITE(12)(X(I),Y(I), I=1,NPLOT)
             READ(1) IGSUM
             WRITE(12) IGSUM
             READ(1) (FDATA(I), I=1,1024 )
             WRITE(12) (FDATA(I), I=1,1024)
             REWIND 1
             WRITE(15,3000) RUNUM
      C******* END OF JOB CONTROL SECTION ******
             IF(CONTRL(28))300,300,259
      259    DID(50)=666666
             DO 270 I=1,49
      270    DID(I)=0
             WRITE(12) DID
             WRITE(15,1200)
             WRITE(6,1200)
             CALL RUNBN(12)
             NGO =3
      C******* THIS EXIT RESTARTS JOB ******
             CALL LINK('IGPHASE1')
      300    IF(CONTRL(1)-3)350,301,350
      C******* P4 CHAN. 2 RECYCLE SECTION ******
      301    IF(SW2)303,303,302
      302    NP4C=2
             NGO=4
             WRITE(15,3010)
             WRITE(6,2050)
             BAND(1)=1.0
             BAND(2)=2.5
             SW2=0
      C******* RECYCLE FROM HERE TO DO P4 CHAN. 2 ******
             CALL LINK('IGPHASE1')
      303    SW2=1
             WRITE(15,3020)
      350    CONTINUE
             IF(CONTRL(20))360,360,355
      C******* END OF RUN CONTROL FOR INDIVIDUAL IG PROCESSING ******
      355    IF(NIG+NR  -CONTRL(14))357,360,360
      357    LAST=START
             IF(CONTRL(1)-3)359,358,359
      358    BAND(1)=.35
             BAND(2)=1.0
      359    CONTINUE
             NPR=0
             MINSAM=1024
             DO 361 I=1,1024
      361    IGSUM(I)=0
             NFIRST=START
             NGO=5
      C******* RECYCLE POINT FOR INDIVIDUAL IG PROCESSING ******
             CALL LINK('IGPHASE1')
      360    CONTINUE
             NGO=2
      C******* RECYCLE POINT FOR AVERAGING ALL IG'S  ******
             CALL LINK('IGPHASE1')
      C******* GRAPH LABEL PRINT SECTION ******
      340    WRITE(6,1090) RUNUM,DID(10)
             WRITE(6,1057) DID(28),(DID(I), I=29,37)
             WRITE(6,1060) DID(3),DID(4),DID(5),FDID(3),FDID(4)
             IF(DID(8)-3)115,116,116
      115    WRITE(6,1070) FDID(1),FDID(2)
      116    WRITE(6,1072) NNPTS,DID(11)
             WRITE(6,1080) FDID(5),FDID(6)
             GO TO(100,190,430),IGO
      C******* FORMAT STATEMENTS ******
      1130   FORMAT(/'     ENTER ''TAKE'' OR ''SKIP''    ')
      1140   FORMAT(/'  INVALID RESPONSE'/'  RETYPE')
      1150   FORMAT(/ ,2X,I10,' TRANSFORMED')
      1160   FORMAT(A4)
      1200   FORMAT(/'     END OF DATA RECORD WRITTEN ON TAPE')
      1010   FORMAT(I10,2X,11I3)
      1045   FORMAT('1',42X,'FOURIER TRANSFORM OF INTERFEROMETER DATA'/ 52X,'FR
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

          2EQUENCY PLOT'//1X)
      1047  FORMAT('0',52X, 'RELATIVE FREQUENCY')
      1050  FORMAT('1',42X,'FOURIER TRANSFORM OF INTERFEROMETER DATA'/ 52X,'WA
          2VELENGTH PLOT'//1X)
      1051  FORMAT('0',52X, 'WAVELENGTH (MICRONS)')
      1057  FORMAT('0',25X,'SOURCE INSTRUMENT - ',A4,19X, 'DATA SCURCE - ',9A4
          2)
      1060  FORMAT('0',25X,'DATE OF RUN - ', I2,'/', I2,'/', I2,21X,'WAVELENGT
          2H BAND - ',F4.2, ' TO ',F5.2, ' MICRONS' )
      1070  FORMAT('0',25X,'SWEEP TIME SETTING = ',F6.3,' SEC.',11X,'SWEEP LEN
          2GTH SETTING = ',F6.2,' MICRONS')
      1072  FORMAT('0',25X,'NUMBER OF SAMPLES TRANSFORMED = ',I4,7X, 'FIRST IG
          2= NO. ',I4 )
      1080  FORMAT('0',25X,'MEAN SQUARE VALUE OF DATA= ',F7.3,9X,      'MEAN S
          2QUARE VALUE OF TRANSFORM= ',F8.3)
      1090  FORMAT('0',25X,'RUN NUMBER = ',I12,18X,'DATA AVERAGED OVER',I4,' S
          2CANS')
      2000  FORMAT('  DATA NOT ON TAPE'/'  SORRY')
      2020  FORMAT('  ID CHECK FLAG'/'  INCORRECT IN RUN ',I3)
      2025  FORMAT('  DATA PUNCH CALLED'/'  LOAD CARDS'/'  ENTER EOB')
      2030  FORMAT(I10,I7)
      2040  FORMAT(10F8.3)
      2045  FORMAT(80X)
      2050  FORMAT('1')
      2060  FORMAT('0',I3,I11,13(1X,I6,1X),/,10(1X,F8.2,1X),/,24A4,I10)
      2080  FORMAT(10X,I5,10X,F7.2,10X,F8.3)
      2090  FORMAT(/'  RELOAD CONTROL CARDS'/'  ENTER EOB')
      3020  FORMAT(/'    P4 CHAN. 2 PROCESSED.')
      3010  FORMAT(/'    P4 CHAN. 1 PROCESSED.')
      3000  FORMAT(/,2X,I10,' WRITTEN ON OUTPUT TAPE')
      5000  FORMAT('0',5X,'NUM PTS. PLOTTED='I4'.  REL. FREQ. OF UPPER WAVELNG
          2TH LMT. ='I4'.'/'      REL. FREQ. OF LOWER WAVLNGTH. LMT. ='I4'.'
          3'  SCALE FACTOR='F10.3)
           END
```

N IV MODEL 44 PS VERSION 2 DATE 68027

COMMON BLOCK / / MAP SIZE C02F5C

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CCCCC0	IGSUM	C01C20	CONTRL	002C20	DID	002C94
002D5C	NP4C	002D60	SW2	002D64	NR	002D68
002D70	LAST	CC2D74	START	0C2D78	BAND	002D7C
C02DA4	NDPNA	002E24	EOFF	0C2E34	NUM	002E38
002E40	SKPP	002E70	NOFST	002E74	SEEK	002E94
002E9C	NSAM	C02EA0	NDPN	002EA4	NDPE	002EA8
002EB0	ISAM	C02EB4	NCLK	CC2EDC	IPAR	002EE0
002EE8	ISFRST	002EEC	NLAST	CC2EF0	NPR	002EF4
002F2C	IFINI	002F30	NCHAN	002F34	NPNCH	002F38
002F40	SWEEPT	CC2F44	SWEEPL	002F48	P4SF	002F4C
002F54	NOFSET	002F58				

SCALAR MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATICN	SYMBOL	LOCATION
CC024C	SKIP	000250	BLANKS	CC0254	RUNUM	000258
C00260	LAMLC	C00264	NPTS	000268	NNPTS	00026C
000274	FNUM	CC0278	FNPTS	0C027C	SCALF	000280
C00288	LCWPT	00028C	LPTPLS	000290	LCWPNT	000294
CC029C	IGO	0002A0	FI	CC02A4	J	0002A8
0002B0	NLO	CC02B4	LIM	0002B8	I2	0002BC
0002C4	FNPLOT	C002C8	DLAM	0002CC	FLAM	0002D0
CC02D8	FRE	C002DC	IFRE	CC02E0	FRE2	0002E4
CC02EC	NNN	0002F0	IN	0002F4		

ARRAY MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATICN	SYMBOL	LOCATION
0002F8	X	000AF8	VAL1	0C12F8	VAL2	001304

SUBPROGRAMS CALLED

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CC1310	GRAPH1	C01314	RUNBN	CC1318	LINK	00131C

LABEL MAP

LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
0013D2	42	CC14D8	55	CC1526	57	001584
0015CC	72	C015DA	73	0015F2	74	001610
00166C	83	CC16A6	85	0016B8	90	0016C0
001752	150	CC1798	155	0017AE	160	0017D6
001818	175	CC1826	176	00183E	180	001896
001944	200	CC1960	410	001A52	235	001A78
CC1AF0	240	CC1B16	243	001B5C	250	001BE6
CC1C40	255	001C70	260	001C80	265	001C9E
CC1E2C	270	C01E44	300	001ED4	301	001EEC
C01F66	350	C01F94	355	CC1FA4	357	001FBC
C0200C	361	CC202C	360	002094	340	0020B2
0021B4	1130	CC2234	1140	CC225E	1150	002288
C022B0	1010	C022E2	1045	0022F4	1047	002342
0023B6	1057	C023DC	1060	C0241A	1070	002470
C025CC	1090	C02562	2000	C025A0	2020	0025C6
002632	2040	CC2640	2045	00264E	2050	002658
C02690	2090	0026A6	3020	0026D8	3010	0026FE
00274A						

MEMORY REQUIREMENTS C02878 BYTES

HIGHEST SEVERITY CODE WAS 0

BN EOJ

```
AN IV    MODEL 44  PS           VERSION 2         DATE  68002

      C TRANSF
      C*****************************************************************
      C*          RCUTINE FOR PREPROCESSING DATA TC BE FOURIER          *
      C*          TRANSFORMED BY SUBROUTINE FCRT (FAST FOURIER TRANSFORM.)    *
      C*****************************************************************
            SUBRCUTINE TRANSF(DATA,Y,X,NUMIN,MAXPT,NUMCUT,IHALF,IGRAPH,IPRINT)
            DIMENSICN DATA(2048),S(512),NDATA(6),XLINE(11),YLINE(11),YLIN(6),
           1DUMMY(512)
            PI=3.1415926
            N=NUMIN
      C FIND INTEGER N2-SUCH THAT NUMIN .GT. 2**N2
            M=1
            DO 14 I=1,10
            M=2*M
            IF(M-N)14,13,13
      13    N2=I
            GC TO 15
      14    CONTINUE
            WRITE(6,207)
      207   FORMAT('1ERROR. NUMBER CF POINTS IS GREATER THAN 1024.')
            RETURN
      15    X=0.0
      C REMOVE D.C. VALUE FROM DATA
            SUM=0.0
            DC 40 I=1,N
      40    SUM=SUM+DATA(2*I-1)
            X=SUM/N
            DO 41 I=1,N
      41    DATA(2*I-1)=DATA(2*I-1)-X
      C FIND MAX PT CF DATA , SEE IF TRANSFORM CF CENTRAL HALF CF DATA ABOUT MAX
      C  VALUE CAN BE TAKEN, IF DESIRED
            CALL MAX(DATA,N,K,2)
      42    N=2**N2
      C RATNC IS RATIC OF OPTICAL CENTER TO NUMBER CF TCTAL POINTS TIMES 2*PI
      C FOR NORMAL TRANSFORM IT IS (K/NUMIN)* 2*PI    , FOR IHALF XFORM IT IS PI
            RATNC=2*PI*R/NUMIN
            R=K
            RATNC=2*PI*R/N
            IF(IHALF)26,22,26
      26    IF(K-N/4)29,29,27
      27    IF(K-(3*N/4))28,29,29
      29    WRITE(6,23)
      23    FCRMAT('0THE LCCATION OF THE MAXIMUM VALUE CF THE DATA IS EITHER L
           1OWER THAN N/4 OR HIGHER THAN 3*N/4.',/' THIS FACT PRECLUDES THE US
           3E OF THE MIDDLE HALF CF THE DATA TC BE TRANSFORMED (TO INCREASE SM
           4OOTHING OF THE TRANSFORM).',/' THEREFORE THIS TRANSFORM IS COMPUTE
           5D CN ALL N PCINTS OF THE DATA')
            GO TO 22
      28    N2=N2-1
            N=N/2
            LOWPT=K-N/2
            LPTPLS=LCWPT+N
            L=0
            DO 25 I=LCWPT,LPTPLS
            L=L+1
      25    DATA(2*L-1)=DATA(2*I-1)
            RATNC=PI
      22    Y=0.0
            DC 4 I=1,N
            K=2*I-1
         4  Y=Y+DATA(K)*DATA(K)
            Y=Y/1024
      C PERFCRM TRANSFCRM BY SUBROUTINE FCRT
            CALL FCRT(DATA,N2,S,-1,IFERR)
      C REDUCE NUMBER OF DATA POINTS DUE TO SYMMETRY OF FOURIER TRANSFORM
            N=N/2
            NUMOUT=N
            IF(IFERR)444,445,444
      444   WRITE(6,334)IFERR
      334   FORMAT('1ERRCR NUMBER ',I3,' OCCURRED IN FORT SUBROUTINE. SEE WRIT
           1E UP')
      C CONVERT DATA FRCM REAL/IMAGINARY CCMPONENTS TC AMPLITUDE/PHASE ANGLE STORAG
      445   DO 703 I=1,N
            R=DATA(2*I-1)
      703   DUMMY(I)=SQRT(DATA(2*I)*DATA(2*I)+R*R)/(IHALF+1)
            CALL MAX(DUMMY,N,NN,1)
            MAXPT=NN
```

```
AN IV     MODEL 44  PS              VERSION 2          DATE  68002

                CHG=C.C
                DO 755 I=1,N
                PICHG=0.0
                ZERO=(I-1)*RATNC
                C=DATA(2*I-1)
                T=DATA(2*I)
                DATA(2*I-1)=DUMMY(I)
                IF(C)865,861,866
          861   IF(T)863,862,864
          862   DIV=0.0
                GO TO 754
          863   DIV=-10000.
                GO TO 754
          864   DIV=10000.
                GO TO 754
          865   DIV=T/C
                PICHG=PI
                GO TO 754
          866   DIV=T/C
            754 DATA(2*I)=AMOD((ATAN(DIV)+ZERO-PICHG),2*PI)
          755   CONTINUE
          901   DO 900 I=1,N
          900   DATA(2*I-1)=DUMMY(I)
          C CONVERSION OF DATA NOW COMPLETED
          71    IF(IPRINT)37,38,37
          37    WRITE(6,333)(I,DATA(2*I-1),DATA(2*I),I=1,N)
          333   FORMAT(1X,2(1X,I5,5X,F9.5,5X,F9.5,10X),1X)
          38    X=0.0
                DO 9 I=1,N
                X=X+DATA(2*I-1)*DATA(2*I-1)
             9  CONTINUE
                X=2*X
                IF(IHALF.NE.1) GO TO 230
                X=X*2
          230   IF(IGRAPH)231,2100,231
          231   L=0
                IF(MAXPT-50)235,236,236
          235   LOWPT=1
                LPTPLS=100
                GO TO 237
          236   IF(MAXPT-N+50)238,239,239
          239   LPTPLS=N
                LOWPT=LPTPLS-99
                GO TO 237
          238   LOWPT=MAXPT-50
                LPTPLS=LOWPT+99
          237   DO 233 I=LOWPT,LPTPLS
                L=L+1
                S(L)=I
          233   DUMMY(L)=DATA(2*I-1)
          2100  CONTINUE
                RETURN
                END
```

AN IV MODEL 44 PS VERSION 2 DATE 68002

SCALAR MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000180	N	000184	NUMIN	000188	M	00018C
000194	X	000198	SUM	00019C	K	0001A0
0001A8	IHALF	0001AC	LOWPT	0001B0	LPTPLS	000184
0001BC	IFERR	0001C0	NUMOUT	0001C4	NN	0001C8
0001D0	PICHG	0001D4	ZERO	0001D8	C	0001DC
0001E4	IPRINT	0001E8	IGRAPH	0001EC		

ARRAY MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
0001F0	S	0001F4	NDATA	0009F4	XLINE	000A0C
000A64	DUMMY	000A7C				

SUBPROGRAMS CALLED

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
00127C	MAX	001280	FORT	001284	FIXPI=	001288
001290						

LABEL MAP

LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
001436	14	001444	207	001478	15	0014BA
001536	42	00157E	26	001614	27	001632
001674	28	0017A0	25	001802	22	001840
0018E0	334	001904	445	00194A	703	001972
001A80	863	001A92	864	001AA4	865	001AB6
001ACC	755	001B20	901	001B32	900	001B3A
001B7C	333	001BE8	38	001C0A	9	001C42
001C86	235	001CA2	236	001CBC	239	001CD4
001D0E	233	001D4E	2100	001D78		

L MEMORY REQUIREMENTS 001DEC BYTES

R HIGHEST SEVERITY CODE WAS 0

```
AN IV    MODEL 44  PS            VERSION 2          DATE  68002
      C          FCRT, ONE-DIMENSICNAL FINITE CCMPLEX FOURIER TRANSFORM.          FORT
      C                                                                           FORT
            SUBROUTINE FCRT(A,M,S,IFS,IFERR)                                      FORT
      C                                                                           FORT
      C     FOURIER TRANSFORM SUBRCUTINE, PROGRAMMED IN SYSTEM/360,               FORT
      C     BASIC PROGRAMMING SUPPORT, FORTRAN IV. FORM C28-6504                  FORT
      C     THIS DECK SET UP FOR IBSYS ON IBM 7094.                               FORT
      C                                                                           FORT
      C     DOES EITHER FOURIER SYNTHESIS,I.E.,COMPUTES COMPLEX FOURIER SERIES    FORT
      C     GIVEN A VECTOR CF N COMPLEX FOURIER AMPLITUDES,OR, GIVEN A VECTOR     FORT
      C     OF COMPLEX DATA X DOES FOURIER ANALYSIS, CCMPUTING AMPLITUDES.        FORT
      C     A IS A CCMPLEX VECTOR OF LENGTH N=2**M COMPLEX NCS. OR 2*N REAL       FORT
      C     NUMBERS. A IS TO BE SET BY USER.                                      FORT
      C     M  IS AN INTEGER 0.LT.M.LE.13, SET BY USER.                           FORT
      C     S IS A VECTOR S(J)= SIN(2*PI*J/NP ), J=1,2,....,NP/4-1,               FORT
      C     CCMPUTED BY PROGRAM.                                                  FORT
      C     IFS IS A PARAMETER TO BE SET BY USER AS FOLLOWS-                      FORT
      C     IFS=0 TO SET NP=2**M AND SET UP SINE TABLE.                           FORT
      C     IFS=1 TC SET N=NP=2**M, SET UP SIN TABLE, AND DO FOURIER              FORT
      C     SYNTHESIS, REPLACING THE VECTOR A BY                                  FORT
      C                                                                           FORT
      C     X(J)= SUM CVER K=0,N-1 OF A(K)*EXP(2*PI*I/N)**(J*K),                  FORT
      C     J=0,N-1, WHERE I=SQRT(-1)                                             FORT
      C                                                                           FORT
      C     THE  X'S  ARE STCRED WITH   RE X(J) IN CELL  2*J&1                    FORT
      C     AND  IM X(J)  IN CELL  2*J&2  FOR  J=0,1,2,...,N-1.                   FORT
      C     THE  A'S  ARE STCRED IN THE SAME MANNER.                              FORT
      C                                                                           FORT
      C     IFS=-1   TO SET  N=NP=2**M,SET UP SIN TABLE, AND DO FOURIER           FORT
      C     ANALYSIS, TAKING THE INPUT VECTOR A AS X AND                          FORT
      C     REPLACING IT BY THE A  SATISFYING THE ABOVE FOURIER SERIES.           FORT
      C     IFS=&2 TC DO FOURIER SYNTHESIS CNLY, WITH A PRE-CCMPUTED S.           FORT
      C     IFS=-2 TO DO FCURIER ANALYSIS ONLY, WITH A PRE-COMPUTED S.            FORT
      C     IFERR   IS SET BY PROGRAM TO-                                         FORT
      C     =0 IF NC ERROR DETECTED.                                              FORT
      C     =1 IF M IS OUT CF RANGE., OR, WHEN IFS=&2,-2, THE                     FORT
      C     PRE-COMPUTED S TABLE IS NOT LARGE ENOUGH.                             FORT
      C     =-1 WHEN IFS =&1,-1, MEANS ONE IS RECCMPUTING S TABLE                 FORT
      C     UNNECESSARILY.                                                        FORT
      C                                                                           FORT
      C     NCTE- AS STATED ABOVE, THE MAXIMUM VALUE CF M FOR THIS PROGRAM        FORT
      C     ON THE IBM 7094 IS 13. FCR 360 MACHINES HAVING GREATER STORAGE        FORT
      C     CAPACITY, CNE MAY INCREASE THIS LIMIT BY REPLACING 13 IN              FORT
      C     STATEMENT 3 BELOW BY LOG2 N, WHERE N IS THE MAX. NO. OF               FORT
      C     COMPLEX NUMBERS CNE CAN STORE IN HIGH-SPEED CORE.  ONE MUST           FORT
      C     ALSO ADD MCRE  DO  STATEMENTS TO THE BINARY SORT ROUTINE              FORT
      C     FCLLCWING STATEMENT  24  AND CHANGE THE EQUIVALENCE STATEMENTS        FORT
      C     FOR THE  K'S.                                                         FORT
      C                                                                           FORT
            DIMENSICN A(1), S(1), K(14)                                           FORT
            EQUIVALENCE (K(13),K1),(K(12),K2),(K(11),K3),(K(10),K4)               FORT
            EQUIVALENCE (K( 9),K5),(K( 8),K6),(K(7),K7),(K( 6),K8)                FORT
            EQUIVALENCE (K( 5),K9 ),(K( 4),K10),(K( 3),K11),(K( 2),K12)           FORT
            EQUIVALENCE (K( 1),K13),( K(1),N2)                                    FORT
            IF(M)2,2,3                                                            FORT
          3 IF(M-13) 5,5,2
          2 IFERR=1                                                               FORT
          1 RETURN                                                                FORT
          5 IFERR=0                                                               FORT
       1001 N=2**M                                                                FORT
       1002 IF( IABS(IFS) - 1 ) 200,200,10                                        FORT
      C     WE ARE DOING TRANSFORM ONLY. SEE IF PRE-CCMPUTED                      FORT
      C     S TABLE IS SUFFICIENTLY LARGE                                         FORT
         10 IF( N-NP )20,20,12                                                    FORT
         12 IFERR=1                                                               FORT
       1003 GC TC 200                                                             FORT
      C     SCRAMBLE A, BY SANDE'S METHCD                                         FORT
         20 K(1)=2*N                                                              FORT
       1004 DO 22 L=2,M                                                           FORT
         22 K(L)=K(L-1)/2                                                         FORT
       1005 DO 24 L=M,12                                                          FORT
         24 K(L+1)=2                                                              FORT
      C     NOTE EQUIVALENCE OF KL AND K(14-L)                                    FORT
      C     BINARY SORT-                                                          FORT
       1006 LJ=2                                                                  FORT
       1007 DO 30 J1=2,K1,2                                                       FORT
       1008 DO 30 J2=J1,K2,K1                                                     FORT
```

```
AN IV     MODEL 44  PS              VERSION 2          DATE  68002

          1009   DO 30 J3=J2,K3,K2                                          FORT
          1010   DO 30 J4=J3,K4,K3                                          FORT
          1011   DO 30 J5=J4,K5,K4                                          FORT
          1012   DO 30 J6=J5,K6,K5                                          FORT
          1013   DO 30 J7=J6,K7,K6                                          FORT
          1014   DO 30 J8=J7,K8,K7                                          FORT
          1015   DO 30 J9=J8,K9,K8                                          FORT
          1016   DO 30 J10=J9,K10,K9                                        FORT
          1017   DO 30 J11=J10,K11,K10                                      FORT
          1018   DO 30 J12=J11,K12,K11                                      FORT
          1019   DO 30 JI=J12,K13,K12                                       FORT
          1030   IF(IJ-JI)28,30,30                                          FORT
             28  T=A(IJ-1 )                                                 FORT
          1031   A(IJ-1)=A(JI-1)                                            FORT
          1032   A(JI-1)=T                                                  FORT
          1035   T=A(IJ)                                                    FORT
          1036   A(IJ)=A(JI)                                                FORT
          1037   A(JI)=T                                                    FORT
             30  IJ=IJ+2                                                    FORT
          1038   IF(IFS)32,2,36                                             FORT
          C      DOING FOURIER ANALYSIS,SO DIV. BY N AND CONJUGATE.         FORT
             32  FN = N                                                     FORT
          1039   DO 34 I=1,N                                                FORT
          1040   A(2*I-1) = A(2*I-1)/FN                                     FORT
             34  A(2*I)=-A(2*I)/FN                                          FORT
          C      SPECIAL CASE- L=1                                          FORT
             36  DO 40 I=1,N,2                                              FORT
          1041   T = A(2*I-1)                                               FORT
          1042   A(2*I-1) =T + A(2*I+1)                                     FORT
          1043   A(2*I+1)=T-A(2*I+1)                                        FORT
          1044   T=A(2*I)                                                   FORT
          1045   A(2*I) = T + A(2*I+2)                                      FORT
             40  A(2*I+2)= T - A(2*I+2)                                     FORT
          1046   IF(M-1) 2,1   ,50                                          FORT
          C      SET FOR L=2                                                FORT
             50  LEXP1=2                                                    FORT
          C      LEXP1=2**(L-1)                                             FORT
          1047   LEXP=8                                                     FORT
          C      LEXP=2**(L&1)                                              FORT
          1048   NPL= 2**MT                                                 FORT
          C      NPL = NP* 2**-L                                            FORT
             60  DO 130 L=2,M                                               FORT
          C      SPECIAL CASE- J=0                                          FORT
          1049   DO 80 I=2,N2,LEXP                                          FORT
          1050   I1=I + LEXP1                                               FORT
          1051   I2=I1+ LEXP1                                               FORT
          1052   I3 =I2+LEXP1                                               FORT
          1053   T=A(I-1)                                                   FORT
          1054   A(I-1) = T +A(I2-1)                                        FORT
          1055   A(I2-1) = T-A(I2-1)                                        FORT
          1056   T =A(I)                                                    FORT
          1057   A(I) = T+A(I2)                                             FORT
          1058   A(I2) = T-A(I2)                                            FORT
          1059   T= -A(I3)                                                  FORT
          1060   TI = A(I3-1)                                               FORT
          1061   A(I3-1) = A(I1-1) - T                                      FORT
          1062   A(I3   ) = A(I1 )   - TI                                   FORT
          1063   A(I1-1) = A(I1-1) +T                                       FORT
             80  A(I1)    = A(I1   )   +TI                                  FORT
          2063   IF(L-2) 120,120,90
             90  KLAST=N2-LEXP                                             FORT
          1064   JJ=NPL                                                     FORT
          1065   DO 110 J=4,LEXP1,2                                         FORT
          1066   NPJJ=NT-JJ                                                 FORT
          1067   UR=S(NPJJ)                                                 FORT
          1068   UI=S(JJ)                                                   FORT
          1069   ILAST=J+KLAST                                              FORT
          1070   DO 100 I= J,ILAST,LEXP                                     FORT
          1071   I1=I+LEXP1                                                 FORT
          1072   I2=I1+LEXP1                                                FORT
          1073   I3=I2+LEXP1                                                FORT
          1074   T=A(I2-1)*UR-A(I2)*UI                                      FORT
          1075   TI=A(I2-1)*UI+A(I2)*UR                                     FORT
          1076   A(I2-1)=A(I-1)-T                                           FORT
          01077  A(I2  )=A(I   ) - TI                                       FORT
          1078   A(I-1) =A(I-1)+T                                           FORT
          1079   A(I)   =A(I)+TI                                            FORT
```

```
AN IV     MODEL 44  PS              VERSION 2          DATE  68002

      1080  T=-A(I3-1)*UI-A(I3)*UR                                          FORT
      1081  TI=A(I3-1)*UR-A(I3)*UI                                          FORT
      1082  A(I3-1)=A(I1-1)-T                                               FORT
      1083  A(I3)   =A(I1  )-TI                                             FORT
      1084  A(I1-1)=A(I1-1)+T                                               FORT
        100 A(I1)   =A(I1)   +TI                                            FORT
      C     END OF I LOOP                                                   FORT
        110 JJ=JJ+NPL                                                       FORT
      C     END OF J LOOP                                                   FORT
        120 LEXP1=2*LEXP1                                                   FORT
      1085  LEXP = 2*LEXP                                                   FORT
        130 NPL=NPL/2                                                       FORT
      C     END OF L LOOP                                                   FORT
        140 IF(IFS)145,2,1                                                  FORT
      C     DOING FOURIER ANALYSIS. REPLACE A BY CONJUGATE.                 FORT
        145 DO 150 I=1,N                                                    FORT
        150 A(2*I) =-A(2*I)                                                 FORT
        160 GO TO 1                                                         FORT
      C     RETURN                                                          FORT
      C     MAKE TABLE OF S(J)=SIN(2*PI*J/NP),J=1,2,....NT-1,NT=NP/4        FORT
        200 NP=N                                                            FORT
      1086  MP=M                                                            FORT
      1087  NT=N/4                                                          FORT
      1088  MT=M-2                                                          FORT
      1089  IF(MT) 260,260,205                                              FORT
        205 THETA=.7853981634                                               FORT
      C     THETA=PI/2**(L&1)     FOR L=1                                   FORT
        210 JSTEP = NT                                                      FORT
      C     JSTEP = 2**( MT-L&1 ) FOR L=1                                   FORT
      1090  JDIF = NT/2                                                     FORT
      C     JDIF = 2**(MT-L)  FOR L=1                                       FORT
      1091  S(JDIF) = SIN(THETA)                                            FORT
      1092  IF (MT-2)260,220,220                                            FORT
        220 DO 250 L=2,MT                                                   FORT
      1093  THETA = THETA/2.                                                FORT
      1094  JSTEP2 = JSTEP                                                  FORT
      1095  JSTEP = JDIF                                                    FORT
      1096  JDIF = JDIF/2                                                   FORT
      1097  S(JDIF)=SIN(THETA)                                              FORT
      1098  JC1=NT-JDIF                                                     FORT
      1099  S(JC1)=COS(THETA)                                               FORT
      2000  JLAST=NT-JSTEP2                                                 FORT
      2001  IF(JLAST-JSTEP)250,230,230                                      FORT
        230 DO 240 J=JSTEP,JLAST,JSTEP                                      FORT
      2101  JC=NT-J                                                         FORT
      2002  JD=J+JDIF                                                       FORT
        240 S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)                                 FORT
        250 CONTINUE                                                        FORT
        260 IF(IFS)20,1,20                                                  FORT
      2005  END                                                             FORT
```

AN IV MODEL 44 PS VERSION 2 DATE 68002

EQUIVALENCE DATA MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000188	K13	000188	N2	000188	K12	00018C
000194	K9	000198	K8	00019C	K7	0001A0
0001A8	K4	0001AC	K3	0001B0	K2	0001B4

SCALAR MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
0001C0	IFERR	0001C4	N	0001C8	IFS	0001CC
0001D4	IJ	0001D8	J1	0001DC	J2	0001E0
0001E8	J5	0001EC	J6	0001F0	J7	0001F4
0001FC	J10	000200	J11	000204	J12	000208
000210	FN	000214	I	000218	LEXP1	00021C
000224	MT	000228	I1	00022C	I2	000230
000238	KLAST	00023C	JJ	000240	J	000244
00024C	UR	000250	UI	000254	ILAST	000258
000260	JSTEP	000264	JDIF	000268	JSTEP2	00026C
000274	JC	000278	JD	00027C		

ARRAY MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000280	S	000284				

SUBPROGRAMS CALLED

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000288	SIN	00028C	COS	000290		

LABEL MAP

LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
00036A	2	00037E	1	00038A	5	000392
0003AC	10	0003BE	12	0003CE	1003	0003DA
0003EC	22	0003F8	1005	000426	24	00042E
00045C	1008	000464	1009	00046C	1010	000474
000484	1013	00048C	1014	000494	1015	00049C
0004AC	1018	0004B4	1019	0004BC	1030	0004C4
0004F0	1032	000504	1035	00050C	1036	000514
000524	1038	00061E	32	000630	1039	000654
00067C	36	00069C	1041	0006A8	1042	0006C4
0006DC	1045	0006E4	40	0006F0	1046	00070E
000730	1048	000738	60	000746	1049	00074E
000766	1052	000772	1053	00077E	1054	00079A
0007BE	1057	0007C6	1058	0007D2	1059	0007DE
00080C	1062	000818	1063	000824	80	000830
00085E	1064	00086A	1065	000872	1066	00087E
0008A6	1069	0008BA	1070	0008C6	1071	0008CE
0008E6	1074	0008F2	1075	00091E	1076	000934
000958	1079	000964	1080	000970	1081	000994
0009C4	1084	0009D0	100	0009DC	110	0009FA
000A24	130	000A30	140	000A56	145	000A68
000AA4	200	000AAA	1086	000AB2	1087	000ABA
000ADA	205	000AE6	210	000AF2	1090	000AFA
000B28	220	000B38	1093	000B44	1094	000B54
000B64	1097	000B74	1098	000B92	1099	000B9E
000BC4	230	000BD4	2101	000BDC	2002	000BE8
000C60	260	000C76				

L MEMORY REQUIREMENTS 000CD8 BYTES

R HIGHEST SEVERITY CODE WAS 0
ABN EOJ

```
 OBJCODE    ADDR    STMT      SOURCE STATEMENT

                      1  *          LOGOUT SENDS KEYBOARD MESSAGES TO LINE PRINTER
                      2  *          LOGIN RETURNS KEYBOARD OUTPUT TO KEYBOARD
                      3  SLOG    START   0
                      4          ENTRY   LOGOUT
D3D6C7D6E4E340        5          DC      CL7'LOGOUT '
08                    6          DC      X'8'
            00008     7          USING   *,15
5040  0024  00024     8  LOGOUT  ST      4,36(13)
4840  04A8  004A8     9          LH      4,X'4A8'(0)    LOAD TYPR ADDR. IN 4
4040  F03C  00044    10          STH     4,SAVE         SAVE IT
4840  04C8  004C8    11          LH      4,X'4C8'(0)    LOAD PRTR ADDR. IN 4
4040  04A8  004A8    12          STH     4,X'4A8'(0)    STORE PRTR. ADDR. IN TYPR. LOC.
584D  0024  00024    13          L       4,36(13)
92FF  D00C  0000C    14          MVI     12(13),X'FF'
07FE                 15          BCR     15,14
                     16          ENTRY   LOGIN
D3D6C7C9D5           17          DC      CL5'LOGIN'
06                   18          DC      X'6'
            0002C    19          USING   *,15
5040  0024  00024    20  LOGIN   ST      4,36(13)
4840  F018  00044    21          LH      4,SAVE
4040  04A8  004A8    22          STH     4,X'4A8'(0)    RESTORE TYPR. ADDR.IN TYPR. LOC.
584D  0024  00024    23          L       4,36(13)
92FF  D00C  0000C    24          MVI     12(13),X'FF'
07FE                 25          BCR     15,14
                     26  SAVE    DS      1F
                   + 27   LTORG
                     28          END
```

```
LIST        MODULE   IGPART01
LIST        MODULE   IGPART2
LIST        MODULE   IGPART3
LIST      MODULE  SLOG
LIST        MODULE    MAX
LIST        MODULE   UNPK08
LIST        MODULE   UNPKID
LIST        MODULE   FORT
LIST        MODULE   TRANSF
LIST        PHASE  IGPHASE1,S
LIST         INCLUDE   IGPART01,L
LIST         INCLUDE   UNPKID,L
LIST         INCLUDE   UNPK08,L
LIST         INCLUDE   SLOG,L
LIST    AUTOLINK     RUNBN
LIST    AUTOLINK     IBCOM=
LIST    AUTOLINK     PLOT12
LIST    AUTOLINK     LINK
LIST    AUTOLINK     FIXPI=
LIST    AUTOLINK     UNITAB=
LIST    AUTOLINK     RCBORG=
LIST    AUTOLINK     USEROPT
LIST        PHASE  IGPHASE2,S
LIST        INCLUDE   IGPART2,L

ERROR   KA02I    ESD         0000 0001    MAIN44=   0 000000 004200

LIST        INCLUDE   TRANSF,L
LIST        INCLUDE   FORT,L
LIST        INCLUDE   MAX,L
LIST    AUTOLINK     IBCOM=
LIST    AUTOLINK     LINK
LIST    AUTOLINK     FIXPI=
LIST    AUTOLINK     SQRT
LIST    AUTOLINK     ATAN
LIST    AUTOLINK     SIN
LIST    AUTOLINK     RCBORG=
LIST    AUTOLINK     USEROPT
LIST    AUTOLINK     UNITAB=
LIST        PHASE  IGPHASE3,S
LIST        INCLUDE   IGPART3,L

ERROR   KA02I    ESD         0000 0001    MAIN44=   0 000000 004200

LIST    AUTOLINK     IBCOM=
LIST    AUTOLINK     GRAPH1
LIST    AUTOLINK     RUNBN
LIST    AUTOLINK     LINK
LIST    AUTOLINK     RCBORG=
LIST    AUTOLINK     USEROPT
LIST    AUTOLINK     UNITAB=
LIST    AUTOLINK     NOR
LIST    AUTOLINK     NND
LIST    AUTOLINK     MOMPL
LIST    AUTOLINK     FIXPI=
LIST    AUTOLINK     AMAX1
LIST        ENTRY
```

LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
			COMMON		004200	002F5C
IGPHASE1						
007160	00EE9F	381	CSECT	MAIN44=	007160	007160
			* ENTRY	MAIN44	007160	
			CSECT	UNPID	00B1D8	00B1D8
			ENTRY	UNPKID	00B1E0	
			CSECT	UNP8	00B2B0	00B2B0
			ENTRY	UNPK08	00B2B8	
			CSECT	SLOG	00B380	00B380
			ENTRY	LOGOUT	00B388	
			ENTRY	LOGIN	00B3AC	
			CSECT	TAPEIO	00B3C8	00B3C8
			ENTRY	RUNBN	00B4CC	
			ENTRY	NOTE	00B424	
			ENTRY	POINT	00B44E	
			ENTRY	READBN	00B3D0	
			* ENTRY	ENDFYL	00B478	
			* ENTRY	REED	00B3D0	
			* ENTRY	WRITBN	00B3FA	
			* ENTRY	WRYTE	00B3FA	
			* ENTRY	UNLOAD	00B4CC	
			* ENTRY	REWYND	00B4A2	
			* ENTRY	WTMBN	00B478	
			* ENTRY	NOTEBN	00B424	
			* ENTRY	POINBN	00B44E	
			* ENTRY	REWBN	00B4A2	
			CSECT	BOAIBCOM	00B670	00B670
			ENTRY	IBCOM=	00B670	
			* ENTRY	ADCON=	00B72C	
			ENTRY	FIRSTIM	00C8B0	
			CSECT	PLOT12=	00DC48	00DC48
			ENTRY	PLOT12	00DC48	
			CSECT	BOAOVLY	00E5E8	00E5E8
			ENTRY	LINK	00E5F0	
			* ENTRY	LOAD	00E600	
			CSECT	BOAFIXPI	00E6E0	00E6E0
			ENTRY	FIXPI=	00E6E8	
			CSECT	BOAUNITB	00E7C0	00E7C0
			ENTRY	UNITAB=	00E7C0	
			CSECT	BOAFIOCS	00E848	00E848
			ENTRY	RCBORG=	00EE70	
			ENTRY	BUFORG=	00EE6C	
			ENTRY	FIOCS=	00E848	
			* ENTRY	VDIOCS=	00EE74	
			* ENTRY	FIOCD=	00E882	
			CSECT	BOAUOPT	00EE98	00EE98
			ENTRY	USEROPT	00EE98	
IGPHASE2						
007160	00D41F	426	CSECT	MAIN44=	007160	007160
			* ENTRY	MAIN44	007160	
			CSECT	TRANSF=	0074F8	0074F8
			ENTRY	TRANSF	0074F8	
			CSECT	FORT=	0092E8	0092E8
			ENTRY	FORT	0092E8	
			CSECT	MAX=	009FF8	009FF8
			ENTRY	MAX	009FF8	
			CSECT	BOAIBCOM	00A260	00A260
			ENTRY	IBCOM=	00A260	
			* ENTRY	ADCON=	00A31C	
			ENTRY	FIRSTIM	00B4A0	
			CSECT	BOAOVLY	00C838	00C838

LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
			ENTRY	LINK	00C840	
			* ENTRY	LOAD	00C850	
			CSECT	BOAFIXPI	00C930	00C930
			ENTRY	FIXPI=	00C938	
			CSECT	BOASSQRT	00CA10	00CA10
			ENTRY	SQRT	00CA16	
			CSECT	BOASATN2	00CAC0	00CAC0
			ENTRY	ATAN	00CAE0	
			* ENTRY	ATAN2	00CAC6	
			CSECT	BOASSCN	00CC38	00CC38
			ENTRY	SIN	00CC58	
			ENTRY	COS	00CC3C	
			CSECT	BOAFIOCS	00CD40	00CD40
			ENTRY	RCBORG=	00D368	
			ENTRY	BUFORG=	00D364	
			ENTRY	FIOCS=	00CD40	
			* ENTRY	VDIOCS=	00D36C	
			* ENTRY	FIOCD=	00CD7A	
			CSECT	BOAUOPT	00D390	00D390
			ENTRY	USEROPT	00D390	
			CSECT	BOAUNITB	00D398	00D398
			ENTRY	UNITAB=	00D398	
IGPHASE3						
007160	00F9AF	462	CSECT	MAIN44=	007160	007160
			* ENTRY	MAIN44	007160	
			CSECT	BOAIBCOM	0099D8	0099D8
			ENTRY	IBCOM=	0099D8	
			* ENTRY	ADCON=	009A94	
			ENTRY	FIRSTIM	00AC18	
			CSECT	GRAPH1=	00BFB0	00BFB0
			ENTRY	GRAPH1	00BFB0	
			CSECT	TAPEIO	00EC78	00EC78
			ENTRY	RUNBN	00ED7C	
			* ENTRY	ENDFYL	00ED28	
			* ENTRY	NOTE	00ECD4	
			* ENTRY	REED	00EC80	
			* ENTRY	WRITBN	00ECAA	
			* ENTRY	READBN	00EC80	
			* ENTRY	WRYTE	00ECAA	
			* ENTRY	POINT	00ECFE	
			* ENTRY	UNLOAD	00ED7C	
			* ENTRY	REWYND	00ED52	
			* ENTRY	WTMBN	00ED28	
			* ENTRY	NOTEBN	00ECD4	
			* ENTRY	POINBN	00ECFE	
			* ENTRY	REWBN	00ED52	
			CSECT	BOAOVLY	00EF20	00EF20
			ENTRY	LINK	00EF28	
			* ENTRY	LOAD	00EF38	
			CSECT	BOAFIOCS	00F018	00F018
			ENTRY	RCBORG=	00F640	
			ENTRY	BUFORG=	00F63C	
			ENTRY	FIOCS=	00F018	
			* ENTRY	VDIOCS=	00F644	
			* ENTRY	FIOCD=	00F052	
			CSECT	BOAUOPT	00F668	00F668
			ENTRY	USEROPT	00F668	
			CSECT	BOAUNITB	00F670	00F670
			ENTRY	UNITAB=	00F670	
			CSECT	NOR1	00F6F8	00F6F8
			ENTRY	NOR	00F6FC	
			CSECT	NND1	00F728	00F728

LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
			ENTRY	NND	00F72C	
			CSECT	MOMP1	00F758	00F758
			ENTRY	MOMPL	00F75E	
			CSECT	BOAFIXPI	00F7B0	00F7B0
			ENTRY	FIXPI=	00F7B8	
			CSECT	BOAFMAXR	00F890	00F890
			ENTRY	AMAX1	00F8E2	
			ENTRY	AMIN1	00F908	
			* ENTRY	MAX1	00F896	
			* ENTRY	MIN1	00F8BC	

4.

```
N IV    MODEL 44  PS            VERSION 2          DATE  68027

      C IGFILER
      C*****************************************************************
      C*        LABORATORY FOR AGRICULTURAL REMOTE SENSING             *
      C*         INTERFEROGRAM DATA PROCESSING SYSTEM                  *
      C*         INTERFEROGRAM FILE CONTROL PROGRAM                    *
      C*****************************************************************
            INTEGER EDIT,PLOT,PNCH,BOTH,S,BLANKS,YES,ID(50),IN(10),SKPP,ADD,DE
           2LE,REPL,OLF,EODR(50),FILE(3),IGDATA(1024),LIT(3,3),QUIT,GRAF,PRNT
            INTEGER CONT(13),LIST(400,2),IDL(100,2),NINFIL(3),FTCC(22)
            INTEGER IDP(100),EOTR(22)
            LOGICAL GETLST
            REAL FID(10),FDATA(1024),X(100),Y(100),DATOUT(6)
            EQUIVALENCE(ID(16),FID(1))
            EQUIVALENCE(FDATA(1),IGDATA(1))
            COMMON FDATA,IC
            DATA IRED,EDIT,PLOT,PNCH,BOTH,S,M,BLANKS,YES,NO/'REA ','EDI ','PLO
           2 ','PNC ','BOT ','S   ','M   ','    ','YE  ','NO  '/
            DATA LIT/'NEW ','IG F','ILE ','OLD ','IG F','ILE ','NEW ','IG T','
           2APE '/
            DATA ADD,DELE,REPL/'ADD ','DELE','REPL'/
            DATA OLF,NEF,NIG/'OLF ','NEF ','NIG '/
            DATA QUIT,GRAF,PRNT/'STO ','GRA ','PRN '/
      C*
      C******* INITIALIZATION ******
      C*
      1     CALL LOAD('SUBS')
            WRITE(6,1000)
            IDIM=100
            GETLST=.FALSE.
            DO 3 I=1,49
      3     EODR(I)=0
            DO 4 I=1,12
      4     CONT(I)=0
            CONT(1)=1
            CONT(13)=1
            DO 5 I=1,21
      5     EOTR(I)=0
            EOTR(22)=333333
            IEDIT=0
            EODR(50)=666666
            WRITE(15,1001)
            NUM=0
            IREAD=0
            INUT=9
            GO TO 4501
      10    WRITE(15,1010)
            IN(1)=BLANKS
      40    READ(15,2121) IN(1)
            IF(IN(1)-YES)42,45,42
      42    IF(IN(1)-NO)44,50,44
      44    WRITE(15,2045)
            GO TO 40
      C*
      C******* TABLE OF CONTENTS READ SECTION **********************************
      C*
      45    NUM=0
            IREAD=0
      43    INC=BLANKS
            WRITE(15,2240)
            READ(15,2130) INC
            INUT=0
            IF(INC.EQ.OLF) INUT=9
            IF(INC.EQ.NEF) INUT=8
            IF(INC.EQ.NIG) INUT=12
            IF(INUT.EQ.8.OR.INUT.EQ.9.OR.INUT.EQ.11) GO TO 4501
            WRITE(15,2045)
            GO TO 43
      4501  REWIND INUT
            INU1=1
            IF(INUT.EQ.9) INU1=2
            IF(INUT.EQ.12) INU1=3
            IUT1=INUT-7
            IUT2=INUT-8
            FILE(1)=LIT(1,INU1)
            FILE(2)=LIT(2,INU1)
            FILE(3)=LIT(3,INU1)
            IF(IREAD.NE.0) GO TO 87
```

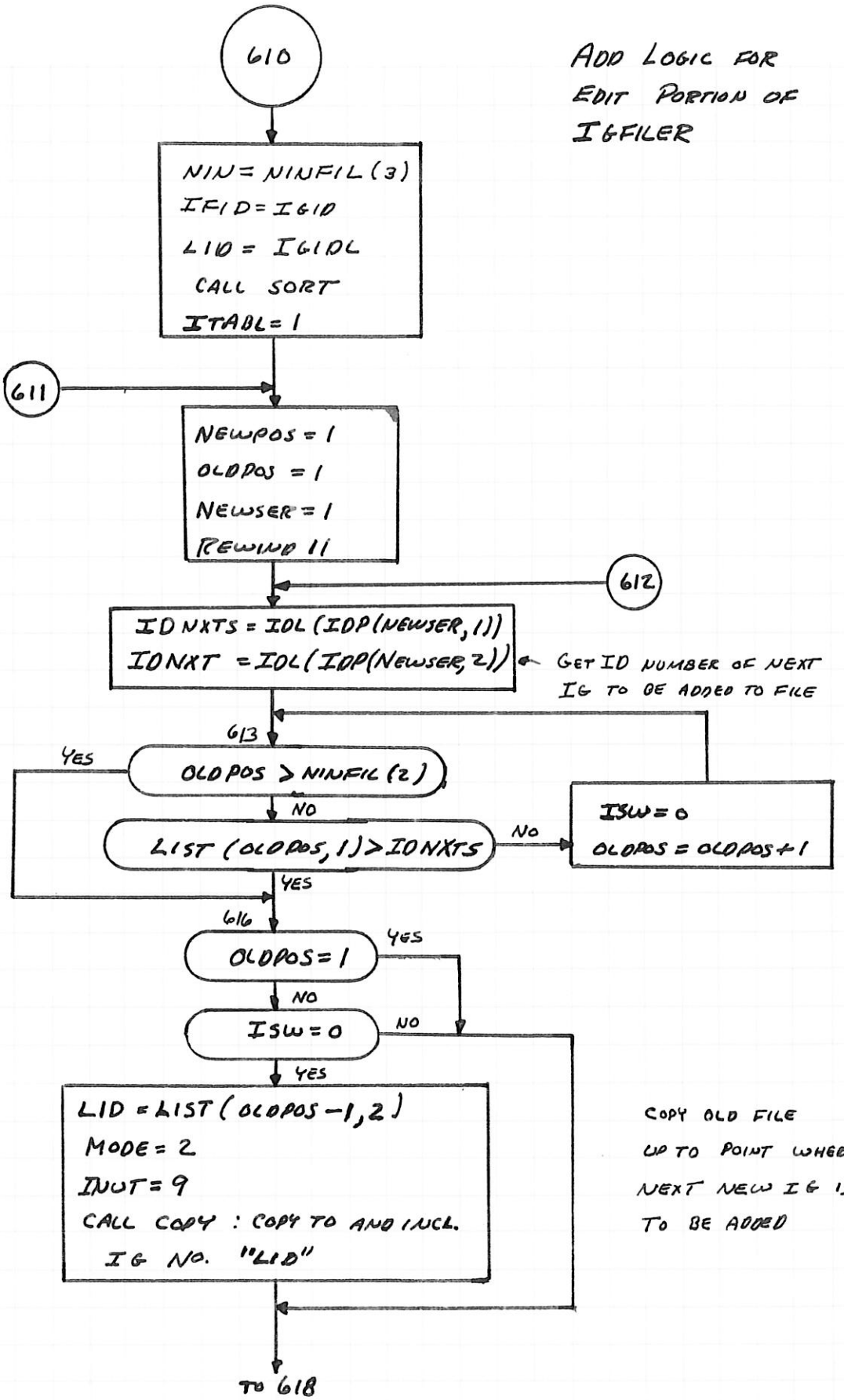
```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

          IF(INUT.EQ.12) GO TO 4502
          WRITE(6,1015) FILE,IUT2
          GO TO 4601
      4502 WRITE(6,1017)
          GO TO 46
      4601 READ(INUT,END=940,ERR=950) FTCC
      4604 IF(FTCC(22).EQ.999999) GO TO 4602
          IF(FTCC(22).EQ.333333) GO TO 49
          ID(2)=FTCC(2)
          GO TO 520
      4602 NUM=NUM+1
          IF(INU1.EQ.1) GO TO 4603
          LIST(NUM,1)=FTCC(1)
          LIST(NUM,2)=FTCC(2)
          NINFIL(2)=NUM
      4603 WRITE(6,1020) NUM,(FTCC(I), I=2,21)
          GO TO 4601
      46  READ(INUT,END=940,ERR=950) ID
          IF(ID(50)-999999)47,48,47
      47  IF(ID(50)-666666)520,49,520
      49  REWIND INUT
          IF(.NOT.GETLST) GO TO 10
          GO TO 600
      48  NNPTS=ID(14)*2
          NUM=NUM+1
          NINFIL(3)=NUM
          IDL(NUM,1)=ID(1)
          IDL(NUM,2)=ID(2)
          READ(INUT,ERR=950) (FDATA(I), I=1,NNPTS)
          READ(INUT,ERR=950) IGDATA
          READ(INUT,ERR=950) FDATA
          WRITE(6,1020) NUM,ID(2),( ID(I), I=29,48)
          GO TO 46
      50  REWIND 9
      C*
      C******* READ MODE SELECT SECTION ****************************************
      C*
      51  WRITE(15,2040)
      52  IN(1)=BLANKS
          READ(15,2130) IN(1)
          IF(IN(1)-IRED)54,60,54
      54  IF(IN(1)-EDIT)56,400,56
      56  IF(IN(1).EQ.QUIT) GO TO 930
          WRITE(15,2045)
          GO TO 52
      60  WRITE(15,2050)
      62  READ(15,2130) IN(1)
          IPLOT=0
          IPRNT=0
          IPNCH=0
          IGRAF=0
          IF(IN(1).EQ.PLOT) IPLOT=1
          IF(IN(1).EQ.PNCH) IPNCH=1
          IF(IN(1).EQ.GRAF)IGRAF=1
          IF(IN(1).EQ.PRNT) IPRNT=1
          IF(IPLOT+IPNCH+IGRAF+IPRNT)68,68,70
      68  WRITE(15,2045)
          GO TO 62
      70  WRITE(15,2055)
          IN(1)=BLANKS
      77  READ(15,2120) IN(1)
          IF(IN(1)-S)78,84,78
      78  IF(IN(1)-M)80,82,80
      80  WRITE(15,2045)
          GO TO 77
      82  WRITE(15,2060)
          READ(15,2140) IGID
          WRITE(15,2065)
          READ(15,2140)  IGIDL
          MULT=1
          GO TO 86
      84  WRITE(15,2070)
          READ(15,2140) IGID
          MULT=0
      86  NRED=0
      C*
      C******* IG SELECT AND READ SECTION ***************************************
```

```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      C*
            IREAD=1
            ISGO=0
            GO TO 43
      87    IF(INUT.EQ.12) GO TO 89
      88    READ(INUT,END=940,ERR=950) FTOC
            IF(FTOC(22).EQ.999999) GO TO 88
            IF(FTOC(22).EQ.333333) GO TO 89
            ID(2)= FTOC(2)
            GO TO 520
      89    READ(INUT,END=940,ERR=950) ID
            NPTS=ID(14)
            NNPTS=NPTS*2
            IF(ID(50).EQ.999999) GO TO 90
            IF(ID(50).EQ.666666) GO TO 500
            GO TO 520
      90    NRED=NRED+1
            IF(MULT.NE.0.AND.ISGO.NE.0) GO TO 94
            IF(ID(2)-IGID)92,93,92
      92    SKPP=1
            GO TO 96
      93    ISGO=1
      94    SKPP=0
            NLOOK=0
      96    READ(INUT,ERR=950) (FDATA(I), I=1,NNPTS)
            IF(SKPP)100,100,97
      97    READ(INUT,ERR=950) IGDATA
            READ(INUT,ERR=950) FDATA
            GO TO 89
      C*
      C***** PLOT SECTION ***************************************************
      C*
      100   IF(IPLOT)200,200,101
      101   INT=NPTS/100
            I1=0
            DO 110 I=1,NPTS,INT
            I1=I1+1
            X(I1)=FDATA(2*I-1)
      110   Y(I1)=FDATA(2*I)
            WRITE(6,1050)
            CALL LOAD('GRAPHER')
            CALL GRAPH1(0,X,Y,100,0.0,0.0,1,0,0)
            CALL LOAD('SUBS')
            WRITE(6,1051)
            WRITE(6,1090) ID(2),ID(28)
            WRITE(6,1057) (ID(I), I=29,48)
            WRITE(6,1060) ID(3),ID(4), ID(5),FID(3),FID(4)
            IF(ID(8)-3)115,120,115
      115   WRITE(6,1070) FID(1),FID(2)
      120   NTRAN=2*ID(6)
            WRITE(6,1072) NTRAN, ID(10)
            WRITE(6,1080) FID(5),FID(6)
            WRITE(6,2100)
      200   IF(IPNCH)210,210,201
      C*
      C******  PUNCH SECTION ***************************************************
      C*
      201   WRITE(15,2160)
              PAUSE 3
            WRITE(7,2170) ID(2),NPTS
            DO 205 NN=1,NNPTS,10
            NNN=NN+9
      205   WRITE(7,2175)  ( FDATA(I), I=NN,NNN )
            WRITE(7,2185)
            WRITE(15,2180)
      210   READ(INUT,ERR=950) IGDATA
            IF(IGRAF.EQ.0) GO TO 220
      C*
      C***** GRAPH SECTION **************************************************
      C*
            WRITE(6,3035) ID(10),ID(2)
            WRITE(6,3037)
            WRITE(6,3039)
            CALL PLOT12(IGDATA,CONT,1,ID(12),1)
      220   READ(INUT,ERR=950) FDATA
            IF(IPRNT.EQ.0) GO TO 300
      C*
```

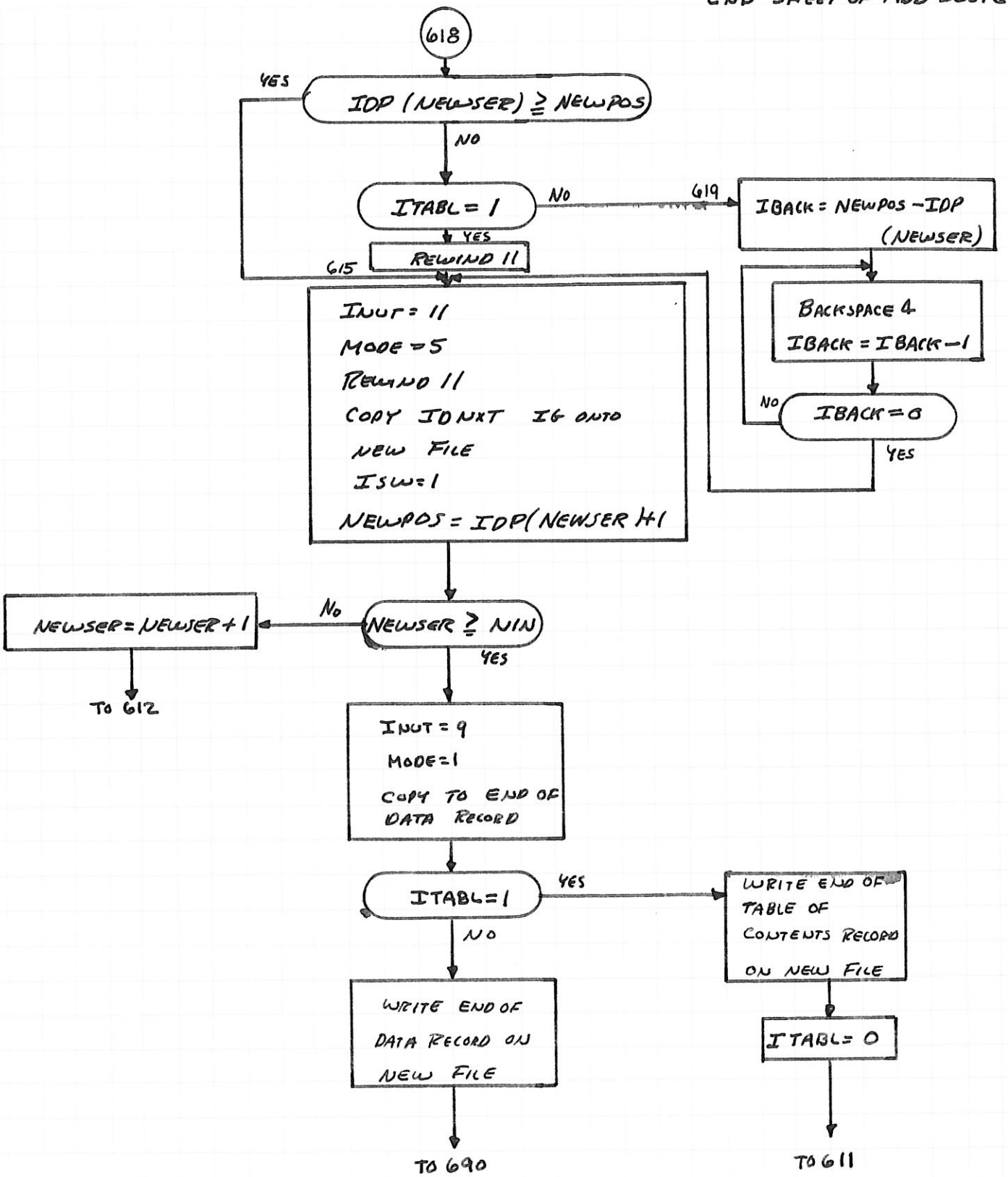
```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      C******* PRINT SECTION ***************************************************
      C*
            WRITE(6,2250) ID(2)
            DO 225 I=1,512,3
            DO 224 I1=1,6
            I2=2*I+I1-2
      224   DATOUT(I1)=FDATA(I2)
      225   WRITE(6,2260) I,DATOUT
      300   IF(MULT)900,900,305
      305   IF(ID(2)-IGIDL)89,900,89
      C*
      C******* EDIT SECTION *****************************************************
      C******* EDIT MODE SELECT ******
      C*
      400   WRITE(15,2190)
            IREAD=0
            IADD=0
            IDELE=0
            IEDIT=1
            IREPL=0
            WRITE(15,2200)
      410   IN(1)=BLANKS
            READ(15,2210) IN(1)
            IF(IN(1)-ADD)415,430,415
      415   IF(IN(1)-DELE)420,435,420
      420   IF(IN(1)-REPL)425,440,425
      425   WRITE(15,2045)
            GO TO 410
      430   IADD=1
            GO TO 445
      435   IDELE=1
            GO TO 445
      440   IREPL=1
      445   IN(1)=BLANKS
            WRITE(15,2220)
      450   READ(15,2120) IN(1)
            IF(IN(1)-S)455,465,455
      455   IF(IN(1)-M)460,470,460
      460   WRITE(15,2045)
            GO TO 450
      465   IMULT=0
            WRITE(15,2070)
            READ(15,2140) IGID
            IGIDL=IGID
            GO TO 480
      470   IMULT=1
            WRITE(15,2060)
            READ(15,2140) IGID
            WRITE(15,2065)
            READ(15,2140) IGIDL
      480   IF(IDELE.EQ.1) GO TO 600
            GETLST=.TRUE.
            INUT=12
            NUM=0
            GO TO 4501
      C*
      C******* EDIT PROCESSING **************************************************
      C*
      600   REWIND 8
            REWIND 9
            GETLST=.FALSE.
            IF(IDELE.EQ.1) GO TO 601
            REWIND 12
      601   IF(IADD.NE.0) GO TO 610
            IF(IDELE.NE.0) GO TO 620
            IF(IREPL.NE.0) GO TO 650
            STOP
      C*
      C******* ADD SECTION **********
      C*
      610   NIN=NINFIL(3)
            IFID=IGID
            LID=IGIDL
            ITABL=1
            CALL SORT(IDL,IDP,IFID,LID,NIN,IDIM,&900)
      611   NEWPOS=1
            OLDPOS=1
```

```
N IV    MODEL 44  PS            VERSION 2          DATE  68027
              NEWSER=1
              REWIND 12
        612   IDNXTS=IDL(IDP(NEWSER),1)
              IDNXT= IDL(IDP(NEWSER),2)
        613   IF(CLDPOS.GT.NINFIL(2)) GO TO 616
              IF(LIST(CLDPOS,1).GT.IDNXTS) GO TO 616
              CLDPOS=CLDPOS+1
              ISW=0
              GO TO 613
        616   IF(CLDPOS.EQ.1) GO TO 618
              IF(ISW.NE.0) GO TO 618
        614   LID=LIST(CLDPOS-1 ,2)
              MODE=2
              INUT=9
              CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
        618   IF(IDP(NEWSER).GE.NEWPOS) GO TO 615
              IF(ITABL.NE.1) GO TO 619
              REWIND 12
        615   INUT=12
              MODE=5
              CALL COPY(INUT,MODE,ITABL,IDNXT,IDNXT,&900)
              ISW=1
              NEWPOS=IDP(NEWSER)+1
              IF(NEWSER.GE.NIN) GO TO 617
              NEWSER=NEWSER+1
              GO TO 612
        619   IBACK=NEWPOS-IDP(NEWSER)
        6191  DO 6192 I=1,4
              BACKSPACE 12
        6192  CONTINUE
              IBACK=IBACK-1
              IF(IBACK.GT.0) GO TO 6191
              GO TO 615
        617   INUT=9
              MODE=1
              CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
              IF(ITABL.EQ.1) GO TO 621
              WRITE(8) EODR
              WRITE(15,3040)
              GO TO 690
        621   WRITE(8) EOTR
              ITABL=0
              GO TO 611
        C*
        C******* DELETE SECTION **********
        C*
        620   INUT=9
              ITABL=1
              IFID=IGID
        625   MODE=3
              CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
              IF(ITABL.EQ.1) GO TO 627
              READ(INUT,END=940,ERR=950) IGDATA
              READ(INUT,END=940,ERR=950) FDATA
        627   IF(IMULT.EQ.0) GO TO 628
              LID=IGIDL
              CALL SKIP(INUT,ITABL,LID,&900)
        628   MODE=1
              CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
              IF(ITABL.EQ.0) GO TO 630
              ITABL=0
              WRITE(8) EOTR
              GO TO 625
        630   WRITE(8) EODR
              WRITE(15,3045)
              GO TO 690
        C*
        C******* REPLACE SECTION *********
        C*
        650   INUT=9
              ITABL=1
              IFID=IGID
        655   MODE=3
              REWIND 12
              CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
              IF(ITABL.EQ.1) GO TO 660
              READ(INUT,END=940,ERR=950) IGDATA
```

ADD LOGIC FOR EDIT PORTION OF IGFILER



2ND SHEET OF ADD LOGIC



```
N IV    MODEL 44  PS              VERSION 2          DATE  68027
      READ(INUT,END=940,ERR=950) FDATA
660   INUT=12
      MODE=5
      LID=IGIDL
      CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
      INUT=9
      IF(IMULT.EQ.C) GO TO 665
      CALL SKIP(INUT,ITABL,LID,&900)
665   MODE=1
      INUT=9
      CALL COPY(INUT,MODE,ITABL,IFID,LID,&900)
      IF(ITABL.EQ.0) GO TO 670
      WRITE(8) ECTR
      ITABL=0
      GO TO 655
670   WRITE(8) ECDR
      WRITE(15,3050)
690   REWIND 8
      REWIND 9
      IF(IDELE.EQ.1) GO TO 900
      REWIND 11
      GO TO 900
C*
C******* END OF DATA RESPONSE SECTION ************************************
C*
500   WRITE(6,2075)
      WRITE(15,2075)
      WRITE(15,2080)
      REWIND 9
      GO TO 900
C*
C******* CHECK CODE ERROR PROCESSING SECTION ********
C*
520   WRITE(15,2090) ID(2),FILE
      WRITE( 6,2090) ID(2),FILE
      REWIND INUT
      GO TO 900
C*
C******* END OF JOB CONTROL SECTION **************************************
C*
900   WRITE(15,2150)
905   IN(1)=BLANKS
      READ(15,2121) IN(1)
      IF(IN(1)-YES)910,930,910
910   IF(IN(1)-NO)920,10,920
920   WRITE(15,2045)
      GO TO 905
930   IF(IEDIT.EQ.0) GO TO 935
      CALL RUNBN(8)
      IF(IDELE.EQ.1) GO TO 935
      CALL RUNBN(12)
935   CONTINUE
      CALL RUNBN(9)
      STOP
C*
C******* SYSTEM EOF RESPONSE SECTION *************************************
C*
940   WRITE(15,2030) FILE,INU2
      WRITE(6,2030) FILE,INU2
      GO TO 900
C*
C******* PARITY ERROR RESPONSE SECTION ************************************
C*
950   WRITE(15,2035) FILE,INU2
      WRITE(6,2035) FILE,INU2
      GO TO 900
C*
C******* END OF EXEC. STMTS. *********************************************
C*
1000  FORMAT('1           LABORATORY FOR AGRICULTURAL REMOTE SENSING '
     2 // '          DIGITAL DATA SYSTEM ' // '                SPECTROMETE
     3R DATA PROCESSING SYSTEM   '// '              INTERFERCGRAM FILE CON
     3TROL PROGRAM')
1001  FORMAT(/ '  MOUNT INTERFEROGRAM DATA FILE TAPE ON UNIT 181 WITH RI
     2NG OUT.')
1010  FORMAT(/ '  TABLE OF CONTENTS '/'  TYPE ''YES'' OR ''NO''  ')
1015  FORMAT('1',20X,'CONTENTS OF ',3A4,                          'ON UN
```

```
N IV    MODEL 44  PS              VERSION 2          DATE  68027

          2LT 18',I1,//,'0', 5X,'INDEX',10X,'ID NUMBER',25X,'SOURCE DESCRIPTI
          3ON')
     1017  FORMAT('1',20X,'CONTENTS OF NEW INTERFEROGRAM DATA TAPE ON UNIT 18
          32 ',///,6X,'INDEX',10X,'ID NUMBER',25X,'SOURCE DESCRIPTION')
     1020  FORMAT('0', 5X,I4,10X,I10,10X,20A4)
     1050  FORMAT('1',42X,'FOURIER TRANSFORM OF INTERFEROMETER DATA'/ 52X,'WA
          2VELENGTH PLOT'//1X)
     1051  FORMAT('0',52X, 'WAVELENGTH (MICRONS)')
     1057  FORMAT('0',20X,'DATA SOURCE - ',20A4)
     1060  FORMAT('0',20X,'DATE OF RUN - ', I2,'/', I2,'/', I2,26X,'WAVELENGT
          2H BAND - ',F4.2, ' TO ',F5.2, ' MICRONS ' )
     1070  FORMAT('0',20X,'SWEEP TIME SETTING = ',F6.3,' SEC.',16X,'SWEEP LEN
          2GTH SETTING = ',F6.2,' MICRONS')
     1072  FORMAT('0',20X,'NUMBER OF SAMPLES TRANSFORMED = ',I4,12X,'DATA AVE
          2RAGED OVER',I4,' SCANS')
     1080  FORMAT('0',20X,'MEAN SQUARE VALUE OF DATA= ',F7.3,14X,     'MEAN S
          2QUARE VALUE OF TRANSFORM= ',F8.3)
     1090  FORMAT('0',20X,'RUN NUMBER = ',I12,23X,'SOURCE INSTRUMENT - ',A4)
     2030  FORMAT(/'     SYSTEM END OF FILE MARK ENCOUNTERED DURING READ OPERA
          2TION FROM ',3A4, 'ON UNIT 18',I1,/'     AN IG RECORD IS MISSING OR
          3AN ECF HAS ERRONEOUSLY BEEN WRITTEN ON AN IG FILE.')
     2035  FORMAT(/'     SYSTEM DATA TRANSMISSION ERROR HAS OCCURRED DURING RE
          2AD OPERATION FROM ',3A4,'ON UNIT 18',I1,/'     RESTART JOB')
     2040  FORMAT(/'  MODE DESIRED   ENTER ''READ'' OR ''EDIT'' OR ''STOP''
          2')
     2045  FORMAT(/'  INVALID RESPONSE DUMBKOPF'/'  RETYPE')
     2050  FORMAT(/'  SELECT READ MODE.  ENTER ''PLOT'' OR ''PNCH'' OR ''GRAF
          2'' OR ''PRNT''   ')
     2055  FORMAT(/'  SINGLE OR MULTIPLE READ   ENTER ''S'' OR ''M''   ')
     2060  FORMAT(/'  TYPE FIRST IG ID NO. (10 DIGITS)   '  )
     2065  FORMAT(/'  TYPE LAST IG ID NO. (10 DIGITS)   '  )
     2070  FORMAT(/'  TYPE ID NO. (10 DIGITS)   ' )
     2075  FORMAT(/'  END OF IG FILE REACHED')
     2080  FORMAT(/'  DESIRED IG NOT ON THIS TAPE.'/'  SORRY ABOUT THAT.')
     2090  FORMAT(/'  ID CHECK CODE INVALID IN IG NO. ',I10,' ON ',3A4,'/'
          2AN IG PROCESSING SYSTEM ERROR HAS OCCURRED.'/'     SUGGEST REWRITIN
          3G TAPE')
     2100  FORMAT('1')
     2120  FORMAT( A1)
     2121  FORMAT(A2)
     2130  FORMAT(A3)
     2140  FORMAT(I10)
     2150  FORMAT(/'  END OF JOB   ENTER ''YES'' OR ''NO''   ')
     2160  FORMAT(/'    DATA PUNCH CALLED FOR.  LOAD BLNK. CARDS, ENTER EOB')
     2170  FORMAT(I10,I7)
     2175  FORMAT(10F8.3)
     2180  FORMAT(/'  PUNCH CMPT.')
     2185  FORMAT(80X)
     2190  FORMAT(/'  MOUNT TAPE WITH NEW IG DATA ON UNIT 182, RING OUT.'/'
          2MOUNT A BLANK NEW IG FILE TAPE ON UNIT 180, RING IN.')
     2200  FORMAT(/'  WHAT FUNCTION DO YOU WISH TO PERFORM   ENTER ''ADD'' OR
          2 ''DELE'' OR ''REPL''   ')
     2210  FORMAT(A4)
     2220  FORMAT(/'  SINGLE OR MULTIPLE   ENTER ''S'' OR ''M''   ')
     2240  FORMAT(/'  OLD FILE, NEW FILE, OR NEW IG TAPE   ENTER ''OLF'', ''N
          2EF'' OR ''NIG''   ')
     2250  FORMAT('1',20X,'NUMERICAL VALUES OF FOURIER TRANSFORM OUTPUT SPECT
          2RUM'/,'0',20X,'ID NO. = ',I11,/ '0',3X,'RELATIVE FREQUENCY   MAGNI
          3TUDE     PHASE ANGLE (DEG.)   ' )
     2260  FORMAT(10X,I4,5X,3(10X,F8.3,5X,F8.3))
     3035  FORMAT('1                GRAPH OF INTERFEROGRAM AVERAGED OVER ',I4,' SC
          2ANS ',/,'0',11X,'ID NO. = ',I11)
     3037  FORMAT('0SAMPLE NUMBER ',45X,' DATA VALUE ' )
     3039  FORMAT( 9X,'0',19X,'50',17X,'100',17X,'150',17X,'200',17X,'250')
     3040  FORMAT(/'  ADD OPERATION SUCCESSFULLY COMPLETED.')
     3045  FORMAT(/'  DELETE OPERATION SUCCESSFULLY COMPLETED.')
     3050  FORMAT(/'  REPLACE OPERATION SUCCESSFULLY COMPLETED.')
           END
```

N IV MODEL 44 PS VERSION 2 DATE 68027

COMMON BLOCK / / MAP SIZE 0010C8

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000000	IODATA	000000	ID	001000	FID	00103C

SCALAR MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000420	EDIT	000424	PLOT	000428	PNCH	00042C
000434	M	000438	BLANKS	00043C	YES	000440
000448	DELE	00044C	REPL	000450	CLF	000454
00045C	QUIT	000460	GRAF	000464	PRNT	000468
000470	I	000474	IEDIT	000478	NUM	00047C
000484	INC	000488	INU1	00048C	IUT1	000490
000498	IPLOT	00049C	IPRNT	0004A0	IPNCH	0004A4
0004AC	IGIDL	0004B0	MULT	0004B4	NRED	0004B8
0004C0	SKPP	0004C4	NLOOK	0004C8	INT	0004CC
0004D4	NN	0004D8	NNN	0004DC	I2	0004E0
0004E8	IREPL	0004EC	IMULT	0004F0	NIN	0004F4
0004FC	ITABL	000500	NEWPOS	000504	OLDPOS	000508
000510	IDNXT	000514	ISW	000518	MODE	00051C
000524						

ARRAY MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
000528	EODR	000550	FILE	000618	LIT	000624
00067C	IDL	0012FC	NINFIL	00161C	FTOC	001628
001810	X	001868	Y	0019F8	DATOUT	001B88

SUBPROGRAMS CALLED

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
001BA0	IBCOM=	001BA4	GRAPH1	001BA8	PLOT12	001BAC
001BB4	SKIP	001BB8	RUNBN	001BBC		

LABEL MAP

LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
001D0A	3	001D48	4	001D76	5	001DB4
001E4C	42	001E7C	44	001E8C	45	001EAA
001FBE	4502	00209E	4601	0020BE	4604	0020E8
00215C	46	00218A	47	0021FC	49	002214
00237A	51	00238C	52	0023A0	54	0023D8
002416	62	002430	68	00240	70	00250E
002560	80	002570	82	00258E	84	002606
00266A	88	00267C	89	0026D6	90	00273E
0027A6	94	0027B2	96	0027C6	97	002828
00287E	110	0028D6	115	002A04	120	002A30
002AAC	205	002B18	210	002BA8	220	002C42
002DC2	300	002D3E	305	002D4A	400	002D64
002DF4	420	002E04	425	002E14	430	002E32
002E56	445	002E62	450	002E84	455	002EB4
002EE2	470	002F32	480	002FA4	600	002FD4
003C58	611	0030AC	612	0030D8	613	00310C
0031A4	618	003214	615	00324C	619	0032CC
0033C0	617	003336	621	0033C2	620	0033F6
0034A8	628	0034F0	630	00356A	650	0035A6
003668	665	0036F4	670	003776	690	0037AC
003842	900	0038AE	905	0038C8	910	003900
00392E	935	003966	940	003982	950	0039E2
003B0C	1010	003B56	1015	003B90	1017	003BEE
003C7A	1051	003CCC	1057	003CF2	1060	003D14
003DC0	1080	003E12	1090	003E68	2030	003EA2
003FD2	2045	004010	2050	004042	2055	004090
0040FC	2070	00412A	2075	004150	2080	004176
004242	2120	004250	2121	00425A	2130	004264
004278	2160	0042AA	2170	0042EC	2175	0042FA
004322	2190	00432C	2200	0043A6	2210	0043FC
00443C	2250	00448E	2260	004524	3035	004542
0045C6	3040	0045F8	3045	00462A	3050	004660

MEMORY REQUIREMENTS 004730 BYTES

HIGHEST SEVERITY CODE WAS 0

3N EOJ

```
N IV    MODEL 44  PS            VERSION 2          DATE  68027

      C COPY
      C*************************************************************************
      C*        COPY PROGRAM FOR INTERFEROGRAM DATA                            *
      C*        INUT = 9  COPYS FROM UNIT 181 TO 180                           *
      C*        INUT = 12  COPYS FROM UNIT 182 TO 180                          *
      C*        MODE = 1  COPYS FROM PRESENT POSITION OF DATA SET              *
      C*        TO THE END OF DATA RECORD (THIS PROG. DOES NOT COPY            *
      C*        THE EOD RECORD.)                                               *
      C*        MODE = 2  COPYS FROM PRESENT POSITION TO AND INCLUDING         *
      C*        IG NO. 'LID'.                                                  *
      C*        MODE = 3  COPYS FROM PRES. POS. TO BUT NOT INCL.               *
      C*        IG NO. 'IFID'.  ID AND DATA RECORD 1 OF IFID REMAIN IN         *
      C*        CORE.  THE FILE IS POSITIONED BEFORE DATA RECORD 2 OF          *
      C*        IFID WHEN RETURN OCCURS.                                       *
      C*        MODE = 4  ADVANCES INPUT FILE TO IG NO. 'IFID' AND             *
      C*        COPYS UP TO THE END OF DATA RECORD.                            *
      C*        MODE = 5  ADVANCES TO IG NO. 'IFID' AND COPYS TO AND           *
      C*        INCL. 'LID'.                                                   *
      C*        ITABL = 1  COPYS TABLE OF CONTENTS ITEMS.  THE TAPE MUST       *
      C*        BE POSITIONED IN THE TABLE OF CONTENTS REGION WHEN             *
      C*        ITABL IS 1 EXCEPT WHEN INUT=12 IN WHICH CASE DATA BLOCKS       *
      C*        ARE CONVERTED TO TABLE OF CONTENTS ENTRIES AND WRITTEN         *
      C*        ON 180.                                                        *
      C*        ITABL = 0  COPYS IG DATA.  THE TAPE MUST BE POSITIONED         *
      C*        IN THE IG DATA PORTION OF THE TAPE WHEN ITABL IS 0.            *
      C*************************************************************************
            SUBROUTINE COPY(INUT,MODE,ITABL,IFID,LID,*)
            INTEGER ID(50),FILE(3),LIT(3,3),IGDATA(1024),R,C,SKIP,FTOC(22)
            REAL FDATA(1024)
            EQUIVALENCE(FDATA(1),IGDATA(1)),(ID(1),FTOC(1))
            COMMON FDATA,ID
            DATA BLANK,R,C    /'    ','R   ','C   '/
            DATA LIT/'NEW ','IG F','ILE ','OLD ','IG F','ILE ','NEW ','IG T','
           2APE '/
            SKIP=0
      15    INU1=2
            IF(INUT.EQ.12) INU1=3
            FILE(1)=LIT(1,INU1)
            FILE(2)=LIT(2,INU1)
            FILE(3)=LIT(3,INU1)
            INU2=INU1-1
      20    ICOPY=0
      25    IF(.NOT.(ITABL.EQ.1.AND.INU1.EQ.2)) GO TO 26
            READ(9,END=100,ERR=120) FTOC
            IF(FTOC(22).EQ.999999) GO TO 50
            IF(FTOC(22).EQ.333333) GO TO 80
            GO TO 35
      26    READ(INUT,END=100,ERR=120) ID
      34    IF(ID(50).EQ.999999) GO TO 39
            IF(ID(50).EQ.666666) GO TO 80
      35    WRITE(15,1050) ID(2),FILE
      38    IN=BLANK
            READ(15,1090) IN
            IF(IN.EQ.R) RETURN 1
            IF(IN.EQ.C) GO TO 39
            WRITE(15,2000)
            GO TO 38
      39    IF(ITABL.NE.1) GO TO 40
            DO 44 I=3,21
      44    FTOC(I)=ID(I+26)
            FTOC(22)=999999
            GO TO 50
      40    NNPTS=ID(14)*2
            READ(INUT,ERR=120,END=100) (FDATA(I), I=1,NNPTS)
      50    IF(MODE.EQ.1.OR.MODE.EQ.2.OR.MODE.EQ.3.OR.ICOPY.NE.0) GO TO 60
            IF(ID(2).NE.IFID) GO TO 55
            ICOPY=1
            GO TO 60
      55    IF(ITABL.NE.1) GO TO 57
            NNPTS=ID(14)*2
            READ(INUT,ERR=120,END=100) (FDATA(I), I=1,NNPTS)
      57    READ(INUT,ERR=120) IGDATA
            READ(INUT,ERR=120) FDATA
            GO TO 25
      60    IF(MODE.EQ.3.AND.ID(2).EQ.IFID) RETURN
            IF(ITABL.EQ.1) GO TO 65
            WRITE(8) ID
```

```
N IV    MODEL 44  PS            VERSION 2         DATE  68027

      WRITE(8) (FDATA(I), I=1,NNPTS)
      READ(INUT,ERR=120) IGDATA
      WRITE(8) IGDATA
      READ(INUT,ERR=120) FDATA
      WRITE(8) FDATA
      GO TO 70
65    WRITE(8) FTOC
      IF(INUT.NE.11) GO TO 70
      NNPTS=ID(14)*2
      READ(INUT,ERR=120,END=100) (FDATA(I), I=1,NNPTS)
      READ(INUT,ERR=120) IGDATA
      READ(INUT,ERR=120) FDATA
70    IF(MODE.EQ.1.OR.MODE.EQ.4) GO TO 25
      IF(ID(2).EQ.LID) RETURN
      GO TO 25
80    IF(MODE.EQ.1.OR.MODE.EQ.4) RETURN
      WRITE(15,2010) FILE
      RETURN 1
100   WRITE(15,2030) FILE,INU2
      WRITE(6,2030) FILE,INU2
      RETURN 1
120   WRITE(15,2035) FILE,INU2
      WRITE(6,2035) FILE,INU2
      RETURN 1
1050  FORMAT('  ID CHECK CODE INCORRECT IN IG NO. ',I11,' ON ',3A4,
     2  /'  RESTART JOB OR CONTINUE '/'  ENTER ''R'' OR ''C''   '/'  CON
     3TINUE ASSUMES EOD NOT YET REACHED')
1090  FORMAT(A1)
2000  FORMAT(/'  INVALID RESPONSE, RETYPE')
2010  FORMAT(/'  EOD ENCOUNTERED BEFORE SPECIFIED LAST IG COPIED'/'  FRO
     2M ',3A4,'.  SUGGEST CHECKING IG NUMBERS.')
2030  FORMAT(/'    SYSTEM END OF FILE MARK ENCOUNTERED DURING READ OPERA
     2TION FROM ',3A4, 'ON UNIT 18',I1,/'    AN IG RECORD IS MISSING OR
     3AN EOF HAS ERRONEOUSLY BEEN WRITTEN ON AN IG FILE.')
2035  FORMAT(/'    SYSTEM DATA TRANSMISSION ERROR HAS OCCURRED DURING RE
     2AD OPERATION FROM ',3A4,'ON UNIT 18',I1,/'    RESTART JOB')
      END
```

N IV MODEL 44 PS VERSION 2 DATE 68027

COMMON BLOCK / / MAP SIZE 0010C8

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CCCCCC	IGDATA	CCCCCC	ID	001000	FTOC	001000

SCALAR MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CCC188	R	CC018C	C	000190	SKIP	000194
00019C	INU2	CCC1A0	ICOPY	0001A4	ITABL	0001A8
0001B0	NNPTS	CC01B4	MCDE	CC01B8	IFID	0001BC

ARRAY MAP

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
0001C4	LIT	CCC1DO				

SUBPROGRAMS CALLED

LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
0001F4						

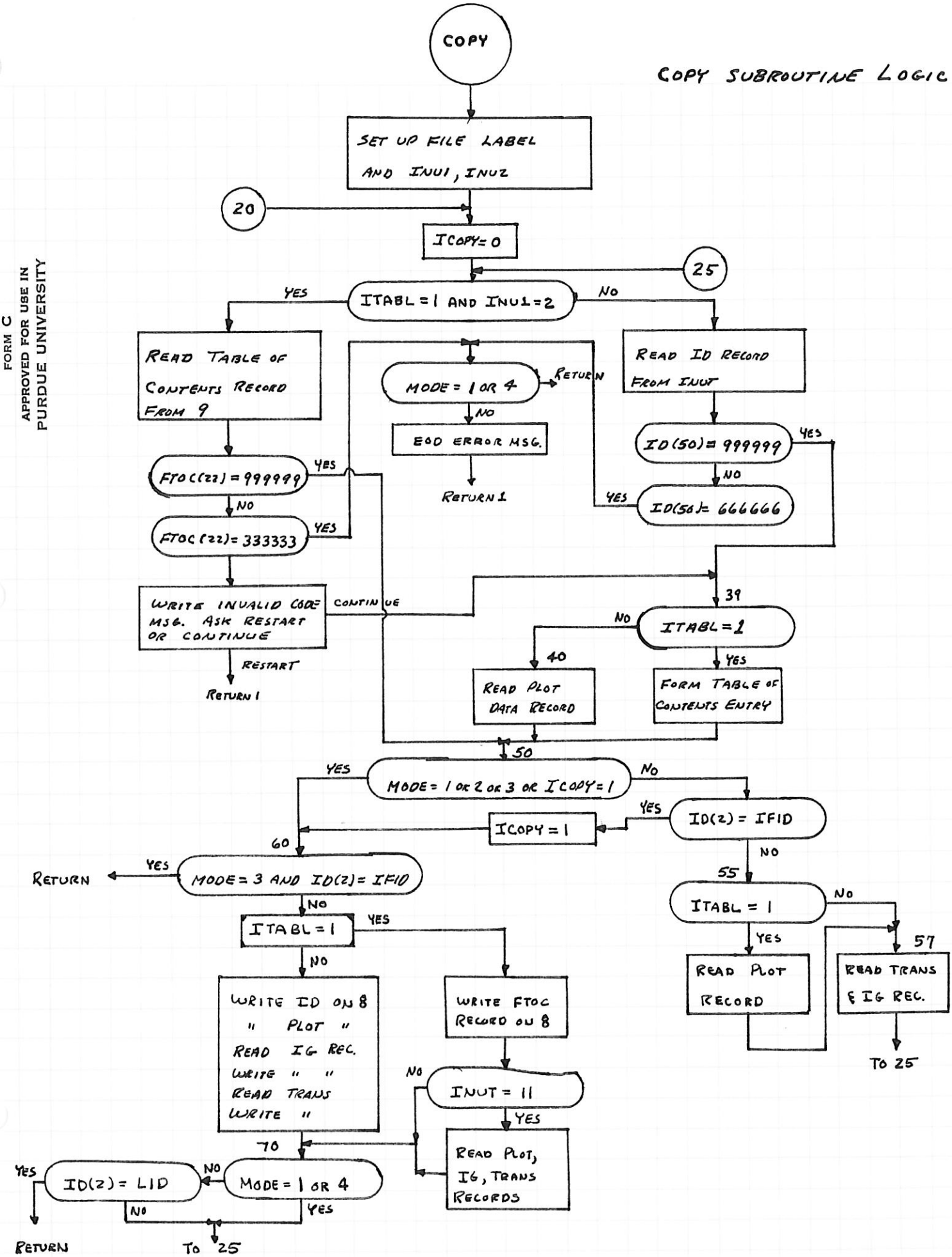
LABEL MAP

LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
CC02CA	20	CC031A	25	000322	26	0003A6
CC03FC	38	CC041C	39	0004A2	44	0004BC
CC055C	55	0005DE	57	000650	60	000696
CC08A0	80	CC08F6	10C	000962	120	0009C2
CC0ABC	2000	CC0AC6	2010	000AEC	2030	000B5A

MEMORY REQUIREMENTS CCCCEC BYTES

HIGHEST SEVERITY CODE WAS C

COPY SUBROUTINE LOGIC



```
N IV     MODEL 44  PS              VERSION 2          DATE  68027

      C SKIP
      C*****************************************************************************
      C*        SKIP ROUTINE FOR INTERFERCGRAM DATA FILES.                         *
      C*        SKIPS FROM PRESENT POSITION TO AND INCL. LID                       *
      C*****************************************************************************
            SUBROUTINE SKIP(INUT,ITABL,LID,*)
            INTEGER ID(50),FILE(3),LIT(3,3),IGDATA(1024),R,C,SKIP,FTOC(22)
            REAL FDATA(1024)
            EQUIVALENCE(FDATA(1),IGDATA(1)),(ID(1),FTOC(1))
            COMMON FDATA,ID
            DATA BLANK,R,C    /'    ','R   ','C   '/
            DATA LIT/'NEW ','IG F','ILE ','OLD ','IG F','ILE ','NEW ','IG T','
           2APE '/
            INU1=2
            IF(INUT.EQ.12) INU1=3
            FILE(1)=LIT(1,INU1)
            FILE(2)=LIT(2,INU1)
            FILE(3)=LIT(3,INU1)
            INU2=INU1-1
      25    IF(.NOT.(ITABL.EQ.1.AND.INU1.EQ.2)) GO TO 26
            READ(9,END=100,ERR=120) FTOC
            IF(FTOC(22).EQ.999999) GO TO 47
            IF(FTOC(22).EQ.333333) GO TO 80
            GO TO 35
      26    READ(INUT,END=100,ERR=120) ID
      34    IF(ID(50).EQ.999999) GO TO 39
            IF(ID(50).EQ.666666) GO TO 80
      35    WRITE(15,1050) ID(2),FILE
      38    IN=BLANK
            READ(15,1090) IN
            IF(IN.EQ.R) RETURN 1
            IF(IN.EQ.C) GO TO 39
            WRITE(15,2000)
            GO TO 38
      39    IF(ITABL.EQ.1) GO TO 47
            NNPTS=ID(14)*2
            READ(INUT) (FDATA(I),I=1,NNPTS)
            READ(INUT,ERR=120) IGDATA
            READ(INUT,ERR=120) FDATA
      47    IF(ID(2).NE.LID) GO TO 25
            RETURN
      80    WRITE(15,2020) FILE
            RETURN 1
      100   WRITE(15,2030) FILE,INU2
            WRITE(6,2030) FILE,INU2
            RETURN 1
      120   WRITE(15,2035) FILE,INU2
            WRITE(6,2035) FILE,INU2
            RETURN 1
      1050  FORMAT(/'  ID CHECK CODE INCORRECT IN IG NO. ',I11,' ON ',3A4,
           2  /'  RESTART JOB OR CONTINUE '/'  ENTER ''R'' OR ''C''   '/'  CON
           3TINUE ASSUMES EOD NOT YET REACHED')
      1090  FORMAT(A1)
      2000  FORMAT(/'  INVALID RESPONSE, RETYPE')
      2020  FORMAT(/'  EOD ENCOUNTERED BEFORE SPECIFIED LAST IG REACHED WHILE
           2SKIPPING ON ',3A4,'.  SUGGEST CHECKING IG NUMBERS.')
      2030  FORMAT(/'    SYSTEM END OF FILE MARK ENCOUNTERED DURING READ OPERA
           2TION FROM ',3A4, 'ON UNIT 18',I1,/'    AN IG RECORD IS MISSING OR
           3AN EOF HAS ERRONEOUSLY BEEN WRITTEN ON AN IG FILE.')
      2035  FORMAT(/'    SYSTEM DATA TRANSMISSION ERROR HAS OCCURRED DURING RE
           2AD OPERATION FROM ',3A4,'ON UNIT 18',I1,/'    RESTART JOB')
            END
```

```
.N IV    MODEL 44  PS            VERSION 2          DATE  68027


                                  COMMON BLOCK /       / MAP SIZE  0010C8
   LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   CCCCCC        IGDATA     CCCCOO        ID         001000        FTOC       001000

                                  SCALAR MAP
   LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   CCC140        R          CCC144        C          000148        INU1       00014C
   000154        ITABL      CCC158        IN         00015C        NNPTS      000160
   CCC168

                                  ARRAY MAP
   LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   00016C        LIT        CCC178

                                  SUBPROGRAMS CALLED
   LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
   CC019C

                                  LABEL MAP
   LOCATION      LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION
   00028A          26       CC030E          34       CC0334          35       000358
   CC04CA          47       CC04B4          80       CC04CE         100       0004FE
   CC058E        1090       CCC658        2000       CC0662        2020       000688
   CC07B4

 MEMORY REQUIREMENTS CC0880 BYTES

 HIGHEST SEVERITY CODE WAS  C
BN ECJ
```

```
AN IV    MODEL 44  PS            VERSION 2         DATE  68002

      C SORT
      C*************************************************************************
      C*        INTERFEROGRAM SORT PROGRAM                                     *
      C*          SORTS LIST OF IG ID NUMBERS IN FIRST COL. OF IDL             *
      C*        (IE. IN IDL(...,1) ) STARTING WITH IFID AND ENDING             *
      C*        AT LID.  RETURNS SORTED LIST OF POINTERS IN IDP.  THE          *
      C*        REQUESTED ID NUMBERS ARE MOVED TO THE START OF IDL             *
      C*        BUT NOT INTERCHANGED.                                          *
      C*************************************************************************
            SUBROUTINE SORT(IDL,IDP,IFID,LID,NUM,IDIM,*)
            INTEGER IDL(IDIM,2),IDP(1),TRANS
            IF(NUM.LT.1) GO TO 150
            DO 30 I=1,NUM
            IF(IDL(I,2).EQ.IFID) GO TO 31
      30    CONTINUE
            IGN=IFID
            GO TO 100
      31    IFRST=I
            DO 32 I=IFRST,NUM
            IF(IDL(I,2).EQ.LID) GO TO 33
      32    CONTINUE
            IGN=LID
            GO TO 100
      33    ILAST=I
            I1=1
            DO 35 I=IFRST,ILAST
            IDL(I1,1)=IDL(I,1)
            IDL(I1,2)=IDL(I,2)
      35    I1=I1+1
            NUM=I1-1
            DO 40 I=1,NUM
      40    IDP(I)=I
            IF(NUM.EQ.1) RETURN
            NUM2=NUM-1
            DO 60 IP=1,NUM2
            NLO=IP+1
            TRANS=0
            DO 50 JP=NLO,NUM
            I=IDP(IP)
            J=IDP(JP)
            IF(IDL(I,1).LE.IDL(J,1)) GO TO 50
            IDS=IDP(IP)
            IDP(IP)=IDP(JP)
            IDP(JP)=IDS
            TRANS=TRANS+1
      50    CONTINUE
            IF(TRANS.EQ.0) RETURN
      60    CONTINUE
            RETURN
      100   WRITE(15,1000) IGN
            WRITE(6,1000) IGN
            RETURN 1
      150   WRITE(15,1010)
            RETURN 1
      1000  FORMAT('0      SORT PROGRAM CANNOT FIND IG NO. ',I10)
      1010  FORMAT(/'     NEW IG TAPE CONTENTS COUNTER = 0. CANNOT CONTINUE.')
            END
```

```
AN IV    MODEL 44  PS            VERSION 2          DATE  68002


                                  SCALAR MAP
    LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
    000118         NUM        000120         I          000124         IFID       000128
    000130         LID        000134         ILAST      000138         I1         00013C
    000144         NLC        000148         TRANS      00014C         JP         000150
    000158

                                  ARRAY MAP
    LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
    00015C         IDP        000160

                              SUBPROGRAMS CALLED
    LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION       SYMBOL     LOCATION
    000164

                                  LABEL MAP
    LOCATION       LABEL      LOCATION       LABEL      LOCATION       LABEL      LOCATION
    00028C           31       0002AC           32       0002E8           33       000308
    0003CE           50       0004E4           60       000510          100       00052E
    00059E         1010       0005D0

L MEMORY REQUIREMENTS 000668 BYTES

R HIGHEST SEVERITY CODE WAS  0
ABN EOJ
```

```
LIST        MODULE  IGFILR00
LIST        MODULE  COPY
LIST        MODULE  SKIP
LIST        MODULE  SORT
LIST        MODULE PLOT12
LIST          MODULE    FLASH
LIST        PHASE  IGFILER0,S,NOAUTO
LIST        INCLUDE  IGFILR00,L
LIST         INCLUDE  IBCOM=,R
LIST          INCLUDE  BUFORG=,R
LIST         INCLUDE  USEROPT,R
LIST          INCLUDE  UNITAB=,R
LIST          INCLUDE  FIXPI=,R
LIST          INCLUDE  AMAX1,R
LIST          INCLUDE  LOAD,R
LIST          INCLUDE   FLASH,L
LIST         PHASE  GRAPHER,*,NOAUTO
LIST         INCLUDE  GRAPH1,R
LIST         INCLUDE  NOR,R
LIST         INCLUDE  NND,R
LIST         INCLUDE  MOMPL,R
LIST         PHASE  SUBS,GRAPHER,NOAUTO
LIST         INCLUDE  PLOT12,L
LIST         INCLUDE  COPY,L
LIST         INCLUDE  RUNBN,R
LIST          INCLUDE  SORT,L
LIST        INCLUDE  SKIP,L
LIST        ENTRY
```

)CORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
			COMMON		004200	0010C8
	IGFILER					
)52C8	00CDEF	511	CSECT	MAIN44=	0052C8	0052C8
			* ENTRY	MAIN44	0052C8	
			CSECT	BOAIBCOM	0099F8	0099F8
			ENTRY	IBCOM=	0099F8	
			* ENTRY	ADCON=	009AB4	
			ENTRY	FIRSTIM	00AC38	
			CSECT	BOAFIOCS	00BFD0	00BFD0
			ENTRY	RCBORG=	00C5F8	
			ENTRY	BUFORG=	00C5F4	
			ENTRY	FIOCS=	00BFD0	
			* ENTRY	VDIOCS=	00C5FC	
			* ENTRY	FIOCD=	00C00A	
			CSECT	BOAUOPT	00C620	00C620
			ENTRY	USEROPT	00C620	
			CSECT	BOAUNITB	00C628	00C628
			ENTRY	UNITAB=	00C628	
			CSECT	BOAFIXPI	00C6B0	00C6B0
			ENTRY	FIXPI=	00C6B8	
			CSECT	BOAFMAXR	00C790	00C790
			ENTRY	AMAX1	00C7E2	
			* ENTRY	MAX1	00C796	
			ENTRY	AMIN1	00C808	
			* ENTRY	MIN1	00C7BC	
			CSECT	BOAOVLY	00C8B0	00C8B0
			ENTRY	LOAD	00C8C8	
			* ENTRY	LINK	00C8B8	
			CSECT	ZOOM	00C9A8	00C9A8
	GRAPHER		* ENTRY	FLASH	00C9AE	
)CDF0	00FB6F	555	CSECT	GRAPH1=	00CDF0	00CDF0
			ENTRY	GRAPH1	00CDF0	
			CSECT	NOR1	00FAB8	00FAB8
			ENTRY	NOR	00FABC	
			CSECT	NND1	00FAE8	00FAE8
			ENTRY	NND	00FAEC	
			CSECT	MOMP1	00FB18	00FB18
			ENTRY	MOMPL	00FB1E	
	SUBS					
)CDF0	00F617	572	CSECT	PLOT12=	00CDF0	00CDF0
			ENTRY	PLOT12	00CDF0	
			CSECT	COPY=	00D798	00D798
			ENTRY	COPY	00D798	
			CSECT	TAPEIO	00E488	00E488
			ENTRY	RUNBN	00E58C	
			* ENTRY	ENDFYL	00E538	
			* ENTRY	NOTE	00E4E4	
			* ENTRY	REED	00E490	
			* ENTRY	WRITBN	00E4BA	
			* ENTRY	READBN	00E490	
			* ENTRY	WRYTE	00E4BA	
			* ENTRY	POINT	00E50E	
			* ENTRY	UNLOAD	00E58C	
			* ENTRY	REWYND	00E562	
			* ENTRY	WTMBN	00E538	
			* ENTRY	NOTEBN	00E4E4	
			* ENTRY	POINBN	00E50E	
			* ENTRY	REWBN	00E562	
			CSECT	SORT=	00E730	00E730
			ENTRY	SORT	00E730	
			CSECT	SKIP=	00ED98	00ED98

OCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
			ENTRY	SKIP	00ED98	

```
IV    MODEL 44  PS            VERSION 2          DATE  68033                    P
      C CALCOMP
      C*************************************************************************
      C*          CALCOMP PLOTTER DATA PREPROCESSING AND FORMATTING PROG.      *
      C*            WRITTEN FOR IBM 7094 COMP. WITH IBSYS OPERATING SYSTEM.    *
      C*************************************************************************
            DIMENSION Y(1026),X(1026),WORK(1024),RUN(2),TITLE(4),XTITLE(4),
           1YTITLE(4),ANAME(8),NR(2)
            DATA ANAME/6HANUTA ,6H3259  , 36H PLEASE USE BLACK INK AND LARGE P
           2EN   /
            DATA RUN/6HRUN NU,6HMBER  /
                              $

            CALL PLOTS(WORK(1),1024,0)
            DIV=20.0
            XINCHS=14.
            YINCHS=7.
            Y1=YINCHS-.25
            Y2=YINCHS-.50
            CALL PLOT(0.0,2.0,-3)
      C
      C
      C  ORDER OF DATA ----
      C      CARD 1 ---   1 NUMERIC CHARACTER -- NUMBER OF SETS OF PLOTS WITH SAME
      C                   TITLE CARDS TO BE PLOTTED
      C      CARD 2 ---   24 ALPHANUMERIC CHARACTERS -- TITLE FOR SET OF PLOTS
      C      CARD 3 ---   24 ALPHANUMERIC CHARACTERS -- TITLE FOR X-AXIS
      C      CARD 4 ---   24 ALPHANUMERIC CHARACTERS -- TITLE FOR Y-AXIS
      C      CARD 5 ---   3 NUMERIC CHARACTERS WITH NUMBER OF RUNS IN THIS SET OF RUNS
      C      CARD 6 --- 1 CHARACTER SPECIFYING  DATA INPUT OPTION (SEE BELOW)
      C      CARD 7 --- 10 CHARACTER RUN NUMBER, 7 CHARACTER NUMBER OF POINTS IN RUN
      C      CARDS 8 AND HIGHER. CONTAIN DATA POINTS, 10 VALUES PER CARD AS PER INPUT
      C
      C       IF CARD 5 .GT. 1, REPEAT CARDS 6 AND FOLLOWING FOR EACH RUN
      C       IF CARD 1 .GT. 1, REPEAT CARDS 2 AND ALL FOLLOWING FOR ALL SETS
      C
      C  THIS PROGRAM HAS THREE OPTIONS FOR FORMAT OF DATA INPUT.
      C  INPUT = 1    - ALL DATA X READ IN, THEN ALL DATA   READ IN. Y DATA STARTS
      C                 ON A NEW CARD IN COL 1.
      C  INPUT = 2    - DATA POINTS READ IN X1,Y1,X2,Y2,...
      C  INPUT = 3    - ONLY Y DATA READ IN. X DATA GENERATED INTERNALLY, PLACING ONE
      C                 Y POINT PER INTERVAL ON X AXIS.
      C
      C
            READ(5,94)NUMSET
      94    FORMAT(I1)
            DO 6 JJ=1,NUMSET
            CALL SYMBOL(-2.0,0.0,.2,ANAME,90.0,48)
            READ(5,4)TITLE,XTITLE,YTITLE
      4     FORMAT(4A6,/4A6,/4A6)
            READ(5,5)NRUNS
      5     FORMAT(I3)
            READ(5,94) INPUT
            DO 6 J=1,NRUNS
            READ(5,1) NR,N
      1     FORMAT(2A6,I5)
            NP1=N+1
            NP2=N+2
            GO TO (101,102,103),INPUT
      101   READ(5,2)(X(I),I=1,N)
            READ(5,2)(Y(I),I=1,N)
            GO TO 500
      102   READ(5,2)(X(I),Y(I),I=1,N)
            GO TO 500
      103   DO 3 I=1,N
      3     X(I)=I
            READ(5,2)(Y(I),I=1,N)
      2     FORMAT(10F8.2)
      500   CALL SCALE(X,XINCHS,N,1,DIV)
            CALL SCALE(Y,YINCHS,N,1,DIV)
            CALL AXIS(0.0,0.0,XTITLE,-24,XINCHS,0.0,X(NP1),X(NP2),DIV)
            CALL AXIS(0.0,0.0,YTITLE,24,YINCHS,90.0,Y(NP1),Y(NP2),DIV)
            CALL SYMBOL(1.0,Y1,.2,TITLE,0.0,24)
            CALL SYMBOL(1.0,Y2,.2,RUN,0.0,12)
            CALL SYMBOL(3.0,Y2,.2,NR,0.0,10)
            CALL LINE(X,Y,N,1,0,0)
            CALL PLOT(20.0,0.0,-3)
      6     CONTINUE
```

```
IV     MODEL 44   PS              VERSION 2          DATE  68033                    P
            CALL PLOT(0,0,999)
            STOP
            END
```

APPENDIX I

Trigger Generator Description

SAMPLING TRIGGER PULSE GENERATOR

In the process of A/D conversion of interferometer data, a sampling trigger pulse derived from the internal clock mechanism of the interferometer must be available in order that the data may be sampled at the appropriate instants. While this interferometer clock signal is in the form of sinusoidal bursts, the sampling trigger pulses must assume the form of negative going spikes. Furthermore, as a consequence of data handling methods, the uniqueness of the initial augmented cycle in each interferometer clock burst (ref. diag. la) must in some way be preserved in each sampling trigger pulse train. It is therefore the twofold function of the sampling trigger pulse generator to:

a. Output a negative going impulse at the same predesignated point on each cycle of the interferometer clock signal.

b. Output a unique negative going "flag pulse" which may be used to signify the beginning of the clock signal burst, and hence the beginning of the interferogram.

Function a is realized by designating the point on each cycle at which the pulse is desired to occur (in this case the negative going zero crossing), and designing a Schmidt trigger circuit which will fire at that voltage level. Then by differentiating the Schmidt trigger output and clipping the positive spike, the required waveform will be obtained. (ref. diag. 2)

The fulfillment of function b is similarly achieved with the uniqueness factor being preserved in the following manner: (ref. diag. la, b, c, d,). Schmidt trigger 1 fires at each negative going zero crossing of the clock burst. Schmidt trigger 2 fires only once per burst, at the negative pre-peak of the augmented cycle. The resulting clipped - and - combined pulse train appears as waveform ld. Now by using this pulse train to sample the interferometer clock signal itself, values of zero will be obtained at each sampled point throughout the burst with the

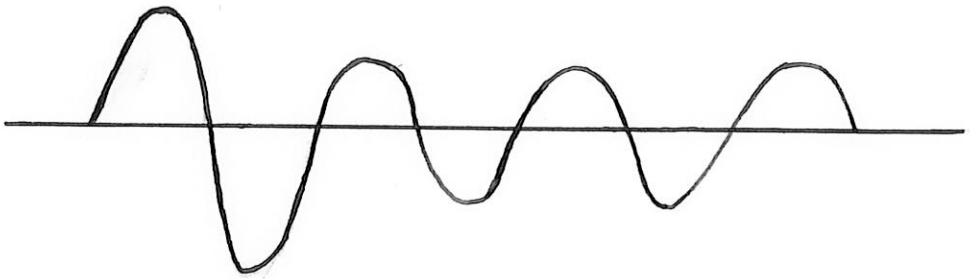
exception of one point, t_1, where a "large" negative value will be found. The presence of this negative value then readily identifies the beginning of each interferogram.

Design Considerations

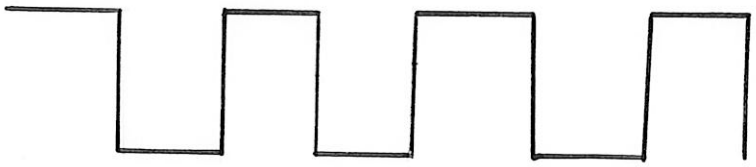
The overall design is straight forward with one exception. Due to the sampling reset time constant inherent in certain projected phases of data analysis, an upper bound is placed upon the pulse train fundamental frequency. Although this limit is not exceeded under conditions of normal operation by the output of either clippers 1 or 2 (ref. diag. 2), it may be exceeded after their superposition (waveform 1d); in which case the "flag" pulse would be skipped because it followed the preceding pulse too closely to effect a system response. To circumvent this situation the combined pulse train is modified by removing the pulse produced by the negative going zero crossing of the augmented cycle. This pulse deletion is accomplished by "anding" the unmodified pulse train (waveform 1d) with the output of a delayed switch circuit (waveform 1e) triggered by the pulse to be eliminated. The required sampling trigger pulse train is then obtained at the output of the "and" gate (waveform 1f).

Construction

The circuitry for the sampling trigger pulse generator was first constructed as a single unit. Later however to facilitate incorporation into the existing A/D conversion system, the circuit subgroups which were available commercially from Raytheon as printed circuit modules were utilized with interfacing where required.



A. INTERFEROMETER CLOCK SIGNAL

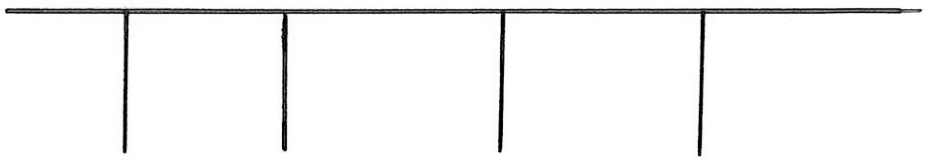


B. SCHMIDT TRIGGER 1 OUTPUT

C. SCHMIDT TRIGGER 2 OUTPUT

D. COMBINED CLIPPER OUTPUT

E. DELAYED SWITCH OUTPUT



F. SAMPLING TRIGGER PULSE OUTLET

DIAGRAM 1. SUBCIRCUIT OUTPUT WAVEFORMS

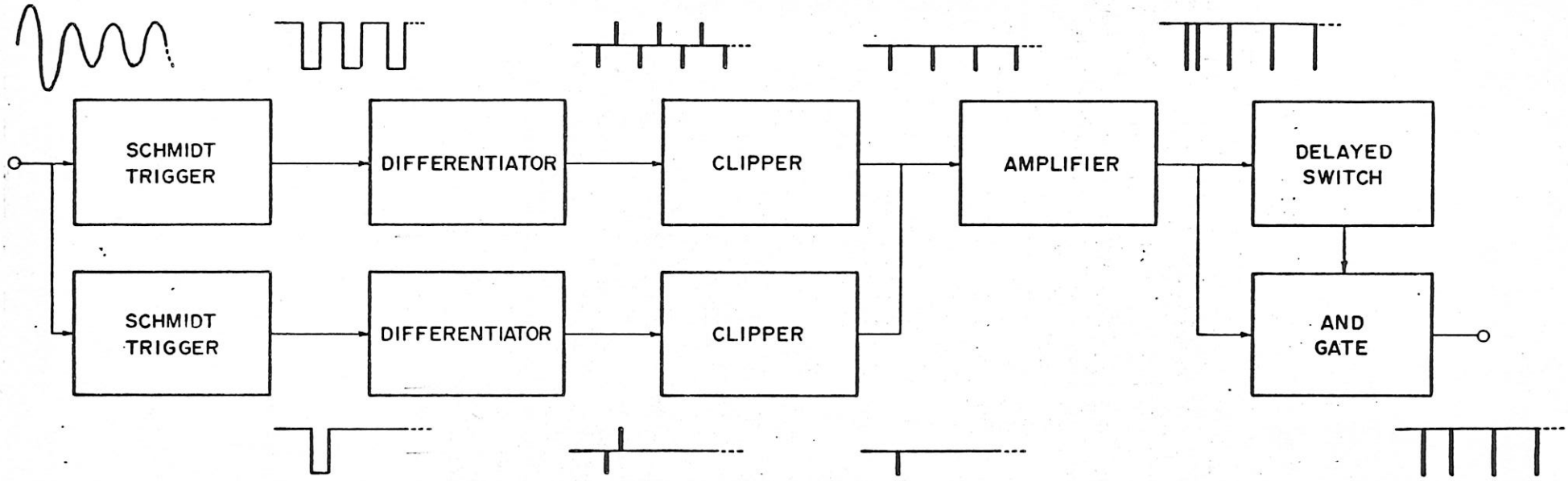


DIAGRAM 2 TRIGGER PULSE GENERATOR

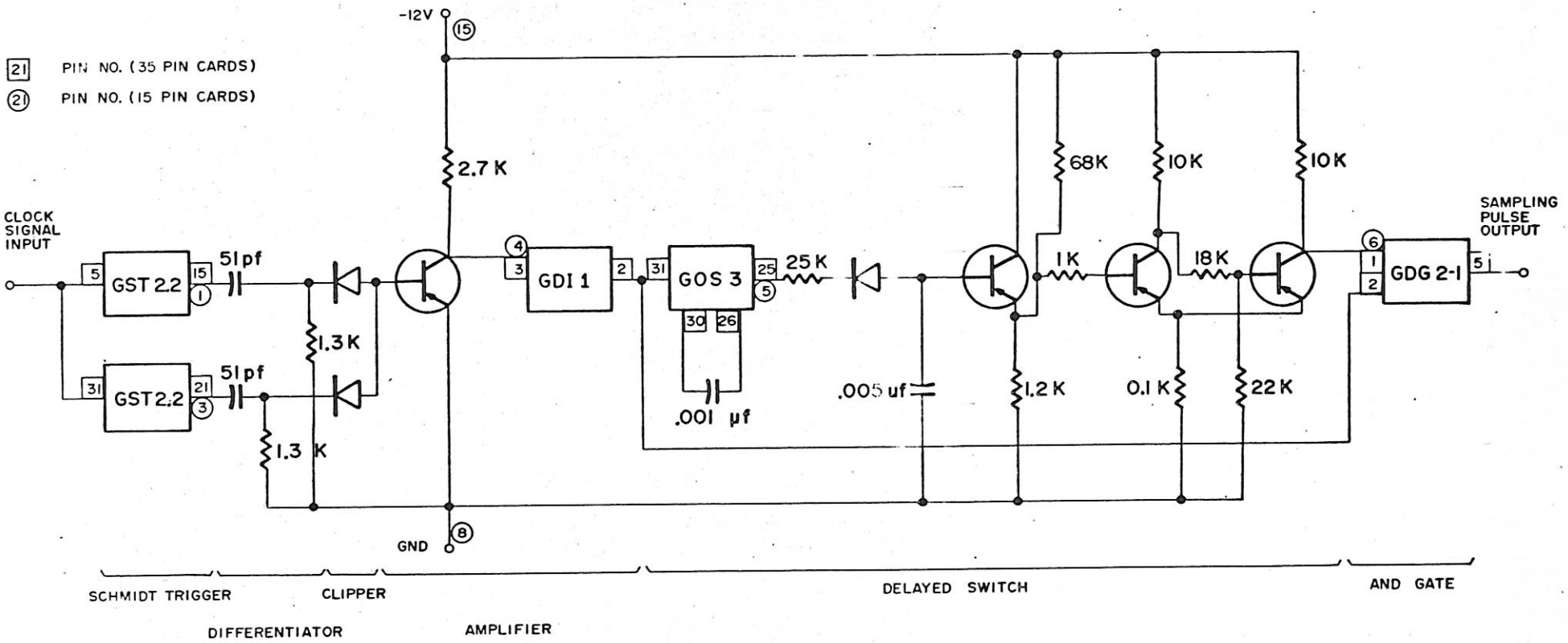


DIAGRAM 3 TRIGGER PULSE GENERATOR CIRCUIT

Adjustment

Adjustment instructions to be included at a later date. Development of the circuit still in progress.

2/2/68.

APPENDIX II

Fourier Transform Program Information

IARS INFORMATION NOTE 061367

A NOTE ON THE USE OF THE FAST FOURIER TRANSFORM

by H. Merrill

The Cooley-Tukey algorithm for computation of the Fourier Transform first appeared in Mathematics of Computation[1] in April, 1965, and has had a tremendous impact on data processing wherein the Fourier transform is a useful instrument. Not only does the algorithm produce a fifty-fold reduction in computation time (the time is proportional to $N \log_2 N$ rather than the normal dependence on N^2), it also produces a more accurate transformation by reducing the number of additions required, and the resulting round off error is proportional to $\log_2 N$ rather than N.

The essence of the algorithm was apparently first developed in the Physics Department of Purdue University in 1942[2] for real valued functions and designed for use on hand calculators, but was never extended to complex functions and digital machines until Cooley-Tukey, unaware of the previous work, developed the algorithm for complex functions of three variables. In the fall of 1966, Cooley himself created a FORTRAN subroutine for implementing the one-dimensional algorithm, and this program is documented under the SHARE Library, Number 3465[3]. A copy of this document is also filed in the IARS Program Library, DH 0004, and it is this program which is discussed herein. The only significant limitation of the algorithm is the requirement that the number of points to be transformed be equal to some power of two, but this limitation can be eliminated.

Three points are to be mentioned herein: (a) the technique when the number of points M is not equal to a power of two; (b) resolution of the resulting transform; and (c) phase angle correction. As will be seen in the third part, the difference between the forward and inverse transforms is essentially the change of a sign in the phase angle calculation, and an appropriate amplitude constant.

The notation used herein is that f(t) is the function (assumed in the time domain for convenience) to be transformed and $F(\omega)$ is the resulting transform, which may be either the forward or inverse transform. The forward transform is defined by

$$F(\omega) = \int f(t) \exp(-j\omega t)\, dt$$

where f(t) may be complex.

a. Number of Points. The basic algorithm requires the number of data points to be a power of two. It is possible to avoid this limitation, if one is careful. If M points are to be transformed, where M is not a power of two, compute the average value of the M points, and subtract this value from each point (i.e., remove the dc value). After removing the dc value, pad the upper end of the data array with zeroes until the number of points is N, the next highest power of two. The desired transform can be then taken on the N points. It is improper to simply pad zeroes to increase the number of points to a power of two, because the algorithm essentially removes the dc value of the waveform prior to transformation, and the zeroes padded on the upper end of the data array would appear as a negative step function with value equal to the dc value of the first M points, resulting in a transform modulated by a $\frac{\text{Sin X}}{\text{X}}$ function.

b. Resolution in the Transform Domain. The algorithm essentially assumes the time-bandwidth product is unity, from which the scaling of the axis in the transform domain is such that the frequency at the i^{th} point is

$$\omega_i = \frac{i}{N}\, \omega_s$$

where i = point number in transform domain (=1, 2, ..., $\frac{N}{2}$)

N = number of points transformed (power of two)

ω_s = sampling rate of time-domain waveform, or

$\omega_s = \frac{2\pi}{t_s}$ where t_s is the sampling period time.

Since the transform is symmetric about the $\frac{N^{th}}{2}$ point in the transform domain, the maximum frequency occurs at this point, and is seen to be $\frac{1}{2}\omega_s$, with which Nyquist would certainly agree. Thus the resolution between points in the transform domain is $\frac{\omega_s}{N}$ and a simple method for increasing the resolution is to pad more zeroes to the data array (after removal of the dc value, as in part a.), thus increasing N and the resolution. If the original waveform sample points did not honestly represent the waveform, increasing the number of points in this manner would incorrectly fill in points in the transform domain, but one generally must assume good representation of the sampled waveform as a starting point.

c. Phase angle considerations. The algorithm returns the real and imaginary parts of the transform points, and a phase angle can be computed directly from these values, but very frequently a more meaningful phase angle spectrum can be constructed. Computing the phase angle based on the real and imaginary points (referred to herein as the uncorrected phase angle) essentially assumes the original waveform had its epoch at zero. For the general case, a waveform with epoch at t_o would introduce a linear phase shift ωt_o in its transform, since if

$$f(t) \longrightarrow F(\omega)$$

then

$$f(t-t_o) \longrightarrow F(\omega)\ e^{j\omega t_o}$$

Removal of the linear phase shift tends to make the dispersion of the phase spectrum less, and variations are more easily seen. By applying the above to a shifted pulse, the magnitude of the necessary correction is found to be $\frac{2\pi n_c}{N}$ where n_c is the epoch point of the time waveform. The sign of the correction is dependent on whether the transform being taken is the forward or inverse. For the forward transform the linear phase shift has negative slope, hence the correction term must have positive sign, and vice-versa. A simple

test is to observe the slope of the uncorrected phase shift and choose the opposite sign for the correction term. Since digitally implemented arctangent rountines are normally module π, it has been found advantageous to displace the corrected phase angle by π to avoid jumps from the top to bottom of the phase spectrum plot, and thus center the spectrum about π radians.

The phase angle computation is given by

$$\theta_i = T_{an}^{-1}\left[\frac{Im\{F(\omega_i)\}}{Re\{F(\omega_i)\}}\right] \pm 2\pi\frac{n_c}{N}i + \begin{cases}\pi \text{ if } Re\{F(\omega)\} \text{ is negative}\\ 0 \text{ if } Re\{F(\omega)\} \text{ is positive}\end{cases}$$

where ω_i = radian frequency of i^{th} point

n_c = epoch point of time waveform

N = number of points transformed

Sign determined as mentioned above.

The change of sign compensates for the direction (forward or inverse) of the transformation, and the only other difference the direction introduces is an amplitude constant, which can be sorted out by either direct calibration, or by observing mean square values of both waveform and transform, and using Parseval's theorem to set the constant.

Cooley's SHARE abstract points out methods of simultaneously transforming two real waveforms in one pass, and methods for transformation data sets which exceed core; in both cases the above remarks apply. Gentleman and Sande[4] discuss methods of convolution by Fourier transformation rather than by normal lagged product methods. Cooley has also written an Assembly Level Program for the three dimensional transform[5]. Substantial work in other areas of application is widespread.

The main advantage of the Fast Fourier Transform is its speed of computation, which has brought the transform, always an excellent theoretical tool, into the domain of nearly real-time data processing techniques. Comparison of transform times have shown for a 500 point transform by a reasonably fast

direct integration algorithm required 180 seconds, as compared with three seconds under the Cooley-Tukey algorithm.

A convenient subroutine for computing transforms using the Cooley-Tukey algorithm with the modifications discussed herein has been written and documented by the author under LARS Program Library DH0003.

REFERENCES

1. Cooley, J. W. and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Math. of Comp., vol. 19, no. 90, (April 1965), pp. 297-301.

2. Danielson, G. C. and C. Lanczos, "Some Improvements in Practical Fourier Analysis and Their Application to X-ray Scattering from Liquids", J. Franklin Inst., vol. 233, (April 1942), pp. 365-380 and pp. 435-452.

3. FORT - "Complex Finite Fourier Transform Subroutine", SDA no. 3465, SHARE Distribution Agency, Program Information Dept., IBM Corp., 40 Saw Mill River Rd., Hawthorne, New York 10532.

4. Gentleman, W. M., and G. Sande, "Fast Fourier Transforms--For Fun and Profit", Proceedings of the Fall Joint Computer Conference, San Francisco, California, Nov. 8-10, 1966.

5. HARM - "Harmonic Analysis Subroutine for IBM 7094, SDA no. 3425". SHARE Distribution Agency. (see ref. III)

COMPLEX FINITE FOURIER TRANSFORM SUBROUTINE

October 6, 1966

Direct inquiries to:

James W. Cooley
IBM Watson Research
Center
P. O. Box 218
Yorktown Heights,
New York 10598

Modifications or revisions to this program, as they occur, will be announced in the appropriate Catalog of Programs for IBM Data Processing Systems. When such an announcement occurs, users should order a complete new program from the Program Information Department.

Table of Contents

Program Abstract

PK FORT Complex Finite Fourier Transform subroutine, written in IBM System/360 BPS FORTRAN IV (see form C28-6504). This subroutine computes the Fourier transform, or its inverse, of a one-dimensional array of N complex numbers where N is a power of 2 up to 8192. The time required on the IBM 7094 is .008, .017, .037, .081 and .175 minutes for N = 512. 1024, 2048, 4096 and 8192, respectively.

This program and its documentation were written by an IBM employee. They have been submitted to the Program Information Department for general distribution in the expectation that they may prove useful to other members of the data processing community. The program and its documentation are, essentially, in the author's original form and have not been subjected to any formal testing. IBM only serves as the distribution agency in supplying this program. It is the user's responsibility to determine the usefulness and technical accuracy of the program in his own environment. This program is not part of the IBM product line as are Programming Systems (Type I) and Application Programs (Type II).

Questions concerning the use of the program should be directed to the author or other designated party. Any changes to tbe program will be reflected in the appropriate Catalog of Programs; however, the changes will not be distributed automatically to users.

Brief Program Description

PK FORT - Subroutine for computing complex finite Fourier transforms, programmed in System/360 Basic Programming Support, FORTRAN IV, Form C28-6504.

1) This program computes the complex Fourier series

$$X(j) = \sum_{k=0}^{N-1} A(k) \exp(2\pi ijk/N) \qquad (1)$$

given the complex vector A(k) or computes A(k) given the vector X(j). The time required for one Fourier transform of N complex data points is approximately $98 N \log_2 N \, 10^{-6}$ sec. on the IBM 7094

This program is essentially a revision of HARM, SDA #3425, now distributed by SHARE, differing in that (1) the present version is entirely in Basic FORTRAN IV; (2) inversion, sorting and table set-up operations are done internally, under control of the arguments; and (3) the present program is limited to one-dimensional Fourier transforms.

2) Method - See, "An Algorithm for the Machine Calculation of Complex Fourier Series", by J. W. Cooley and J. W. Tukey, Mathematics of Computation 19, p. 297, April, 1965. The method is also described below in the section, "Detailed Program Description", accompanying this write-up.

3) Restrictions and Range: Two storage cells are required for each complex number, so one must have a block of 2N cells available in core storage. N must be a power of 2 less then or equal to $2^{13} = 8192$.

4) Precision: The program was checked out on the IBM 7094 under the IBSYS system. To test the accuracy of the program, an array of data was set equal to $X(j) = \exp(2\pi ijk/N)$, $j = 0,1,2,\ldots,N-1$, and k fixed. This was transformed by the subroutine to an array A defined by (1). The correct result, $A(k) = 1$, $A(k') = 0$, $k' \neq k$, was compared with the computed result and the maximum difference and the R. M. S. difference were computed and printed out. With $N = 2^{13}$ these were 7×10^{-8} and 10^{-11}, respectively. Results with other parameter values are given in Table 1 of the detailed program description.

5) Program Requirements: No special machine requirements are needed but the sample program supplied requires a clock to give time

estimates.

6) Source Language: IBM System 360 Basic Programming Support FORTRAN IV, Form C28-6504. The cards were punched on the 026 keypunch and the sample program deck is set up for the IBM 7094 with the IBSYS operating system.

7) Program Execution Time: is found to be

$$kN\log_2 N$$

where

$$k = 98 \times 10^{-6} \text{ sec.}$$

8) Check-Out Status: Calculations with a wide range of parameter values were run giving maximum and R. M. S. errors. Initial and final arrays were printed and examined in some cases. Users have reported successful results from tests and data computed from sums of several sinusoidal functions of various frequencies.

Detailed Program Description

Identification

PK FORT, Subroutine for computing a complex finite Fourier transform. Programmed in IBM System/360 Basic Programming Support, FORTRAN IV. Accompanying deck set up for and checked out on IBM 7094, IBSYS system. James W. Cooley, IBM Watson Research Center, P. O. Box 218, Yorktown Heights, New York.

Purpose

Given an integer M and an array of complex Fourier amplitudes,

$$A(k),\ k = 0,1,2,\ldots,N-1\ , \qquad N = 2^M\ ,$$

the subroutine computes the Fourier Series

$$X(j) = \sum_{k=0}^{N-1} A(k)W^{jk}\ , \tag{1}$$

where $W = \exp(2\pi i/N)$ for a particular setting of one of its arguments. For another setting of this argument, it computes the inverse transformation, i.e., taking the input array to be X it computes the array A satisfying (1). The formula for the inverse is given in the section "Usage" below. It will also be shown below how to perform Fourier transforms of large arrays via several passes through the subroutine and how to compute real Fourier series efficiently.

Method - The Basic Algorithm

The algorithm is derived by writing X(j) in (1) as a function of M arguments, these being the contents of the M bit positions in the binary representation of j. Thus, if

$$j = j_{M-1}\cdot 2^{M-1} + \ldots + j_1\cdot 2 + j_0, \qquad j_\nu = 0 \text{ or } 1,$$

we let

$$X(j) = X(j_{M-1}, \ldots, j_1, j_0)\ .$$

Similarly, if

$$k = k_{M-1}\cdot 2^{M-1} + \ldots + k_1\cdot 2 + k_0\ , \qquad k_\nu = 0 \text{ or } 1,$$

we let

$$A(k) = A(k_{M-1}, \ldots, k_1, k_0) .$$

From A(k) we construct a new array $A_0(k)$ by re-ordering the elements of A(k) in such a way that

$$A_0(k_0, k_1, \ldots, k_{M-1}) = A(k_{M-1}, \ldots, k_1, k_0) .$$

This just says that for each k a new index k' is formed by inverting the order of the M bits of k, giving

$$k' = k_0 \cdot 2^{M-1} + \ldots + k_{M-2} \cdot 2 + k_{M-1} .$$

Then $A_0(k') = A(k)$ and the $A_0(k')$ array can be formed by interchanging the k and k' elements of A(k) for each $k < k'$. In a previous version of this program, HARM SDA #3425, this was done after operation of the algorithm. Here, it is done on the initial array and an appropriately modified algorithm is used.

Now, with all this understood, (1) can be written

$$X(j_{M-1}, \ldots, j_1, j_0) = \sum_{k_0=0}^{1} \sum_{k_1=0}^{1} \cdots \sum_{k_{M-1}=0}^{1} A_0(k_0, k_1, \ldots, k_{M-1})$$

$$\times W^{jk_{M-1} \cdot 2^{M-1}} \times \ldots \times W^{jk_1 \cdot 2} W^{jk_0} . \qquad (2)$$

Since $W^{2^M} = W^N = 1$ we have

$$W^{j \cdot 2^{M-1}} = W^{(j_{M-1} \cdot 2^{M-1} + \ldots + j_1 \cdot 2) \cdot 2^{M-1}} W^{j_0 \cdot 2^{M-1}} = W^{j_0 \cdot 2^{M-1}}$$

and, in general,

$$W^{j \cdot 2^{M-L}} = W^{(j_{M-1} \cdot 2^{M-1} + \ldots + j_L \cdot 2^L)\, 2^{M-L}} W^{(j_{L-1} \cdot 2^{L-1} + \ldots + j_0) 2^{M-L}}$$

$$= W^{(j_{L-1} \cdot 2^{L-1} + \ldots + j_0) \cdot 2^{M-L}}$$

We can perform the innermost summation in (2), summing over k_{M-1}, with all other k_ν's fixed, yielding an array depending upon j_0 and

$k_{M-2}, \ldots, k_0$. This new array can be written

$$A_1(k_0, \ldots, k_{M-2}, j_0) = \sum_{k_{M-1}=0}^{1} A_0(k_0, \ldots, k_{M-1}) W^{j_0 k_{M-1} \cdot 2^{M-1}} \qquad (3)$$

Successive arrays $A_2, A_3, \ldots, A_M$, obtained by summing over k_{M-2}, $k_{M-3}, \ldots, k_0$ respectively, are computed. The general formula is

$$A_L(k_0, \ldots, k_{M-L-1}, j_{L-1}, j_{L-2}, \ldots, j_0) =$$

$$\sum_{k_{M-L}=0}^{1} A_{L-1}(k_0, \ldots, k_{M-L-1}, k_{M-L}, j_{L-2}, \ldots, j_0)$$

$$\times W^{(j_{L-1} \cdot 2^{L-1} + \ldots + j_0) k_{M-L} \cdot 2^{M-L}} . \qquad (4)$$

The final array will be the desired X's. The storage indexing convention used here is to let the M arguments of $A_L(k_0, \ldots, j_0)$ be the binary representation of the index of the storage location for $A_L(k_0, \ldots, j_0)$. In this way, each step of the algorithm involves fetching from two storage locations, corresponding to $k_{M-L} = 0, 1$, (all other indices fixed) and putting the results back in the same two locations. This permits overwriting the input array with results, thereby saving storage.

A simple FORTRAN program can now be written by assigning indices as follows: Let J be an integer defined by the contents of the low-order L-2 bits of the index I of $A_L(I)$ and let K be an integer defined by the high-order M-L bits of the index of A_L. Thus, I is defined by the four indices K, j_{L-1}, j_{L-2}, and J as follows:

$$A_L(I) = A_L(\underbrace{k_0, \ldots, k_{M-L-1}}_{K}, j_{L-1}, j_{L-2}, \underbrace{j_{L-3}, \ldots, j_0}_{J}) \qquad (5)$$

where $J = 0, 1, \ldots, 2^{L-2}-1$; $K = 0, 1, \ldots, 2^{M-L}-1$,

and

$$I = K \cdot 2^L + j_{L-1} \cdot 2^{L-1} + j_{L-2} \cdot 2^{L-2} + J . \qquad (6)$$

Now, noting that

$$W^{j_{L-1}\cdot 2^{L-1}k_{M-L}\cdot 2^{M-L}} = (-1)^{j_{L-1}\cdot k_{M-L}}$$

and

$$W^{j_{L-2}\cdot 2^{L-2}k_{M-L}\cdot 2^{M-L}} = i^{j_{L-2}k_{M-L}}$$

a concise description of the algorithm (4), in a pseudo-FORTRAN language can be given:

```
      COMPLEX A,U,T
C     L = 1 IS A SPECIAL CASE
      DO 5 K = 0,2**(M-1)-1
      T = A(K,1)
      A(K,1) = A(K,0)-T
5     A(K,0) = A(K,0)+T
      DO 10 L = 2,M                                        (7)
      DO 10 J = 0,2**(L-2)-1
      U = W**(J*2**(M-L))
      DO 10 K = 0,2**(M-L)-1
      T = A(K,1,0,J)*U
      A(K,1,0,J) = A(K,0,0,J)-T
      A(K,0,0,J) = A(K,0,0,J)+T
      T = A(K,1,1,J)*U*CMPLX(0.,1.)
      A(K,1,1,J) = A(K,0,1,J)-T
10    A(K,0,1,J) = A(K,0,1,J)+T
```

All A's on the right-hand sides of the equations are A_{L-1}'s and on the left sides, they are A_L's. Note that this is written so that the new array A_L over-writes, or replaces, the array A_{L-1}. Thus, a single array, A, is used for the input, temporary storage of the A_L's and the result.

The actual program deviates from the description (7) by making a special case of the index J = 0. In this case, U is equal to 1 and no multiplications are necessary.

Method: Construction of the Sine Table

The sine table,

$$S(j) = \mathrm{Sin}\frac{\pi}{2^{M+1}}\,j, \qquad j=0,1,2,\ldots,2^M-1$$

is set up by calling FORT with the argument IFS equal to 0 or +1. The table is constructed in M passes, indexed L = 1,2,...,M. At the start of the L-th pass all indices having zeros in their right-most M-L bit positions are assumed to have been done and on the L-th pass, we compute all entries having 1 in the M-L-th bit of their indices. That is, given $S(J\cdot 2^{M-L+1})$ we compute $S(J\cdot 2^{M-L+1}+2^{M-L})$ for $J = 0,1,2,\ldots,2^{L-1}-1$. First we set $S(0) = 0$ and $S(2^{M-1}) = \mathrm{Sin}\frac{\pi}{4}$

Then, for each L = 2,3,...,M we compute

$$S(2^{M-L}) = \mathrm{Sin}\frac{\pi}{2}2^{-L}$$

$$S(2^M-2^{M-L}) = \mathrm{Cos}\frac{\pi}{2}2^{-L}$$

using the SIN and COS subroutines. Then, using the trigonometric sum formula, we compute

$$S(J\cdot 2^{M-L+1}+2^{M-L}) = S(J\cdot 2^{M-L+1})\cdot S(2^M-2^{M-L}) + S(2^M-2^{M-L+1})S(2^{M-L}) \qquad (8)$$

for $J = 0,1,\ldots,2^{L-1}-1$. This procedure only adds positive quantities and, therefore, does not amplify rounding errors.

Timing

To count the number of operations required, observe that there are two complex additions for each pass through statement 5 and four for each pass through statement 10. This gives

$$\sum_{K=0}^{2^{M-1}-1} 2 + \sum_{L=2}^{M}\sum_{J=0}^{2^{L-2}-1}\sum_{K=0}^{2^{M-L}-1} 4$$

$$= 2^{M-1}\cdot 2 + \sum_{L=2}^{M} 2^{L-2}\cdot 2^{M-L}\cdot 4 = N + (M-1)N = MN$$

complex additions. There are two complex multiplications for each pass through statement 10, with $J \neq 0$. (Note that $J \neq 0$ excludes the calculation for $L = 2$). This gives

$$\sum_{L=3}^{M} \sum_{J=1}^{2^{L-2}-1} \sum_{K=0}^{2^{M-L}-1} 2 = \sum_{L=3}^{M} (2^{L-2}-1)2^{M-L}.2$$

$$= (M-2)2^{M-1}-(2^{M-1}-2) \approx \frac{N}{2}(M-3)$$

complex multiplications.

Counting two real additions for each complex addition and two real additions and four real multiplications for each complex multiplication, we have

$$3N(\log_2 N-1) \text{ real additions} \qquad (9)$$

and

$$2N(\log_2 N-3) \text{ real multiplications.} \qquad (9')$$

Consequently, if η and μ are the add and multiply times of the computer, arithmetic operations alone will take an amount of time equal to

$$k'N(\log_2 N-C') \qquad (10)$$

where

$$k' = 3\eta + 2\mu$$

$$C' = \frac{3\eta + 6\mu}{k'}.$$

Assuming $\eta = 7$ and $\mu = 4\times10^{-6}$ sec., we get

$$k' = 29$$

$$C' = 1.6$$

Actual operating times on the IBM 7094, with the program compiled by IBSYS, FORTRAN IV, are given in Table 1 (page 25). These satisfy a formula like (7) with k' and C' replaced by

$$k = 98\times10^{-6} \text{ sec.}$$

and

$$C = 0.$$

This shows that about twice as much time is taken for bookkeeping as for arithmetic. Examination of the compiled program shows this to be so. It is also estimated that about 33% of the total operating time could be saved by rather simple machine-language programming. Such a saving would reduce k to about 65×10^{-6} sec. This agrees with Prof. T. Stockham, of MIT, who programmed the algorithm in machine language and got $k = 60\times10^{-6}$ sec. The constant C', which represents the savings due to avoiding multiplication by 1, is seen to be off-set in the program by set-up operations, i.e., bookkeeping outside the innermost loop.

Accuracy

Experience with the above algorithm was reported by a number of people to yield better accuracy than the conventional procedure of accumulate-adding the Fourier series of N terms. This can apparently be explained by noting that if ϵ is the error incurred in one multiply-add, then the error in the direct method would be ϵN. By the present method, however, one adds pairs of the array A_{L-1} to obtain the array A_L. This is done $\log_2 N$ times so the error in the result is expected to be proportional to $\log_2 N$ instead of N as in the direct method. In Table 1, the ratio of maximum error to $\log_2 N$ is shown to be approximately constant and equal to $.5\times10^{-8}$. In tests with random number data, this ratio increases slowly and almost linearly with $\log_2 N$ from 1.66 at $N = 512$ to 2.29 at $N = 8192$.

Usage

1. The Complex Fourier Series: The subroutine FORT accepts, as input, a one-dimensional array of complex Fourier coefficients A(k) or of data X(i). With the complex array A(k), $k = 0,1,\ldots,N-1$, where $N = 2^M$, as input, the subroutine FORT computes

$$X(j) = \sum_{k=0}^{N-1} A(k)W^{jk} \qquad \text{for } j=0,1,2,\ldots,N-1 \quad (11)$$

where $W = \exp(2\pi i/N)$, and replaces the A's by the respective X's. It is to be noted here that W^{jk} is periodic of order N in both j and k and, therefore, both A(k) and X(j) are periodic of order N in their indices. That is, the N-vector X(j), $j=0,1,\ldots,N-1$ defines X(j) for all j, with the property that $X(j) = X(j \bmod (N))$. The range

of indices in summations and in formulas will be omitted when it is to be understood that the relevant variables are periodic of order N in their indices and that the range of an index is over one period. The notation A or X, without argument, will denote the N-element column vector with elements A(k), k = 0,1,...,N-1, or X(j), j = 0,1,...,N-1, respectively.

2. The Finite Fourier Transform: The inversion formula, for calculating the A satisfying (11) when given X, is

$$A(k) = \frac{1}{N}\sum_{j} X(j)W^{-jk}. \tag{12}$$

Taking complex conjugates of both sides, it is seen that if the input array is $\tilde{X}/N$, the output will be $\tilde{A}$. (Tilde denotes complex conjugate.) The calculation of A from X is usually referred to as Fourier analysis, and A is called the Fourier transform of X. As explained below, FORT computes (12) if the argument IFS is negative and computes (11) if IFS is positive.

3. Simultaneous Fourier Analysis of Two Sets of Real Data: The following procedure can be used for obtaining a Fourier analysis of two sets of real data with one reference to FORT. An alternate procedure is given later for doing a Fourier analysis of one set of real data via one pass through FORT with half as many complex data elements. Let X' and X" be two sets of real data and suppose we wish to obtain their Fourier coefficients, A' and A", satisfying

$$X'(j) = \sum_{k} A'(k)W^{jk} \tag{13}$$

$$X''(j) = \sum_{k} A''(k)W^{jk}. \tag{13'}$$

The procedure is to form the complex array

$$X = X' + iX''$$

and, by application of the Fourier transform, compute the coefficients A of

$$X(j) = \sum_{k} A(k)W^{jk}. \tag{14}$$

To derive the formulas for A' and A" first multiply (13') by i and add to and subtract from (13) to get

$$X'(j) \pm iX''(j) = \sum_{k} (A'(k) \pm iA''(k))W^{jk} \tag{15}$$

The complex conjugate of (14) can be written, by replacing the index of summation by k' = N-k,

$$\tilde{X}(j) = \sum_{k} \tilde{A}(k)W^{-jk} = \sum_{k'} \tilde{A}(N-k')W^{jk'}. \tag{16}$$

Equating coefficients of (14) with those of (15) with the "+" sign gives

$$A(k) = A'(k) + iA''(k) \tag{17}$$

and equating coefficients of (16) with those of (15) with the "-" gives

$$\tilde{A}(N-k) = A'(k) - iA''(k). \tag{17'}$$

Solving these, we get

$$A'(k) = \frac{1}{2}(A(k) + \tilde{A}(N-k)) \tag{18}$$

$$A''(k) = \frac{1}{2i}(A(k) - \tilde{A}(N-k)) \tag{18'}$$

For the special cases, k = 0 and N/2 it is worth noting that

$$A'(0) = \mathcal{R}e\,A(0) \qquad A'(N/2) = \mathcal{R}e\,A(N/2) \tag{19}$$

$$A''(0) = \mathcal{I}m\,A(0) \qquad A''(N/2) = \mathcal{I}m\,A(N/2) \tag{19'}$$

We easily see here the symmetry imposed on the Fourier coefficients of real data. If X' were real, then X", and consequently A", would be zero and, from (18') we see that the Fourier coefficients A of X would have the property that $A(k) = \tilde{A}(N-k)$. Therefore, the real and imaginary parts of A would be symmetric and antisymmetric, respectively, about k = N/2. Since X' and X" are real, their Fourier coefficients satisfy the symmetry property

$$A'(k) = \tilde{A}'(N-k) \tag{20}$$

$$A''(k) = \tilde{A}''(N-k) \tag{20'}$$

and only half of each array need be computed and stored. Thus, one requires the same amount of storage for A' and A" as for A or X.

From (19) we see that we can skip the indices k = 0 and N/2 since we already have these results in the real and imaginary parts of A(0) and A(N/2). To summarize:

Step 1. Let X(j) = X'(j) + iX''(j) be the input to FORT and set the argument IFS = -1 or -2.

Step 2. Call FORT. The negative IFS causes a Fourier transform to be computed, replacing the array X by A.

Step 3. Then, for k = 1, 2, ..., (N/2)-1 let

$$A'(k) = \frac{1}{2}(A(k) + \tilde{A}(N-k)) \tag{21}$$

$$A''(k) = \frac{1}{2i}(A(k) - \tilde{A}(N-k)) . \tag{21'}$$

4. Doubling the Capacity of FORT: Here, it will be shown how an analysis of 2N data points can be performed by doing two analyses of N points each. If N is the maximum number of points which can be kept in core storage due to memory limitations, the procedure permits one to do an analysis of 2N points by performing two passes through FORT. Repetition of the procedure with use of bulk storage permits one to successively double the capacity up to the limitations of the bulk storage. It is interesting to note that the formulas for successive doubling of capacity by the method described below is actually a description of the algorithm used in FORT though with a different indexing scheme.

The procedure is to do two analyses by performing two passes through FORT, once with the even-indexed points and once with the odd-indexed points. These yield the coefficients A'(k) and A''(k) of

$$X(2j) = \sum_{k=0}^{N-1} A'(k)W^{jk} \tag{22}$$

$$X(2j+1) = \sum_{k=0}^{N-1} A''(k)W^{jk} \tag{22'}$$

$$j = 0, 1, 2, \ldots, N-1 .$$

We now derive the formulas used to obtain an analysis of the full set of 2N points from the two analyses of N points each.

Instead of powers of W, the 2N-point Fourier series will be in powers of

$$V = \exp(\pi i/N),$$

the first 2N-th root of unity. We write

$$X(j) = \sum_{k=0}^{2N-1} C(k)V^{jk} , \qquad j = 0, 1, 2, \ldots, 2N-1 \tag{23}$$

The even- and odd-indexed X(j)'s can be expressed

$$X(2j) = \sum_{k=0}^{N-1} C(k)V^{2jk} + \sum_{k=N}^{2N-1} C(k)V^{2jk} \tag{24}$$

$$X(2j+1) = \sum_{k=0}^{N-1} C(k)V^{(2j+1)k} + \sum_{k=N}^{2N-1} C(k)V^{(2j+1)k} . \tag{24'}$$

Using the property

$$V^2 = W$$

and replacing the index of summation in the last N terms of both series by

$$k' = k-N$$

we get

$$X(2j) = \sum_{k=0}^{N-1} C(k)W^{jk} + \sum_{k'=0}^{N-1} C(N+k')W^{j(N+k')} \tag{25}$$

$$X(2j+1) = \sum_{k=0}^{N-1} C(k)W^{jk}V^{k} + \sum_{k'=0}^{N-1} C(N+k')W^{j(N+k')}V^{N+k'} . \tag{25'}$$

Since $W^{jN} = 1$ we can collect terms and equate coefficients of (25) and (25') with those of (22) and (22'). This yields

$$A'(k) = C(k) + C(N+k)$$

$$A''(k) = (C(k) - C(N+k))V^{k}.$$

In the latter equation, we used $V^{N} = -1$. Solving for $C(k)$, we have

$$C(k) = \tfrac{1}{2}\ (A'(k) + A''(k)V^{-k}) \qquad (26)$$

$$C(N+k) = \tfrac{1}{2}\ (A'(k) - A''(k)V^{-k}) \qquad (26')$$

for $k = 0, 1, 2, \ldots, N-1$.

5. A Useful Check on Accuracy: If the $X(j)$'s represent a discrete sampling of a continuous function, a useful check that the sampling density is sufficient and that large errors have not occurred during the calculation can easily be programmed in the calculation. The $A'(k)$'s and $A''(k)$'s represent two analyses of the same set of data but at alternating sampling points. The $C(k)$'s are an analysis of the same data at twice the sampling density and, appropriately, are given by (26) as an average of the results from the sparse samplings with one of them phase-shifted by V^{-k} corresponding to the shift of one sample spacing. A useful check that the sampling density is sufficient and that some point or points are not grossly in error is to see that $C(k)$ agrees reasonably well with $A'(k)$ and $A''(k)V^{-k}$. Since $C(k)$ for the lower half of the k's is the average of $A'(k)$ and $A''(k)V^{-k}$, this is equivalent to asking if the latter are close. But the differences are equal to the $C(k)$'s for $k = N, N+1, \ldots, 2N-1$, the new octave of frequences made possible by doubling the sampling density. Therefore, the error criterion is that the $C(k)$'s with $k = N, N+1, \ldots, 2N-1$ must be relatively small in amplitude. These checks may detect deficiencies in the data while a further check, by means of Parseval's theorem is available to check against computer errors during the calculation. Parseval's theorem states that

$$\sum_{j} |X(j)|^2 = N \sum_{k} |A(k)|^2 \ . \qquad (27)$$

Combining these two checks will usually tell one whether to suspect the computer or the data.

6. Fourier Analysis of Real Data: One can, of course, compute Fourier analyses of real data by doing two sets at a time according to the procedure described above, or, if one has only one set to do, one can just set the imaginary part of the input array to zero.

Here, we will describe a more efficient procedure by which one constructs a complex array of only half as many elements, obtains its transform using FORT, and then performs a few simple calculations to obtain the results.

Suppose we have $2N$ real data $Y(j)$, $j = 0, 1, 2, \ldots, 2N-1$. The coefficients of the trigonometric series

$$Y(j) = \frac{1}{2}a_0 + \sum_{k=1}^{N-1} \left(a_k \mathrm{Cos}\frac{\pi jk}{N} + b_k \mathrm{Sin}\frac{\pi jk}{N}\right) + \frac{1}{2}(-1)^j a_N \qquad (28)$$

can easily be derived from the complex coefficients of the Fourier series.

$$Y(j) = \sum_{k=0}^{2N-1} C(k)V^{jk} \qquad (29)$$

where

$$V = \exp(\pi i/N) \ .$$

As we have seen before, $Y(j)$ being real implies the symmetry condition

$$C(k) = \tilde{C}(2N-k) \ . \qquad (30)$$

Therefore, substituting for the $C(k)$'s with $k \geq N$ in (29), we get

$$Y(j) = \sum_{k=0}^{N-1} C(k)V^{jk} + \sum_{k=N}^{2N-1} \tilde{C}(2N-k)V^{jk} \ .$$

Changing the index of summation to $k' = 2N-k$ and using $V^{2N} = 1$ puts the second sum in the form

$$\sum_{k'=1}^{N} \tilde{C}(k')V^{j(2N-k')} = \sum_{k'=1}^{N} \tilde{C}(k')V^{-jk'} \ .$$

This yields

$$Y(j) = C(0) + 2\,\mathrm{Re}\sum_{k=1}^{N-1} C(k)V^{jk} + (-1)^j C(N)$$

$$= C(0) + 2\sum_{k=1}^{N-1}\left(\mathrm{Re}\,C(k)\cos\frac{\pi jk}{N} - \mathrm{Im}\,C(k)\sin\frac{\pi jk}{N}\right) + (-1)^j C(N)\,.$$

Note that (30) implies that C(0) and C(N) must be real. Comparison with the trigonometric series (28) establishes the following identities:

$$\begin{aligned} a_0 &= 2C(0) \\ a_N &= 2C(N) \\ a_k &= 2\,\mathrm{Re}\,C(k) \\ b_k &= -2\,\mathrm{Im}\,C(k), \qquad k = 1,2,\ldots,N-1. \end{aligned} \tag{31}$$

We can compute the C(k)'s by using the method given above for doubling the capacity of FORT. This gives, by equation (26),

$$C(k) = \tfrac{1}{2}\left(A'(k) + A''(k)V^{-k}\right)$$

where A'(K) and A''(k) are the Fourier coefficients of the even- and odd-indexed real data points, respectively,

$$Y(2j) = \sum_{k=0}^{N-1} A'(k)W^{jk}$$

$$Y(2j+1) = \sum_{k=0}^{N-1} A''(k)W^{jk}\,.$$

The A'(k) and A''(k) can be computed by the procedure described in Section 3 above for obtaining Fourier analyses of two sets of real data in one pass through FORT. Here, the real data are Y(2j) and Y(2j+1), and, by the method in Section 3, these are used to form the complex array

$$X(j) = Y(2j) + iY(2j+1) = \sum_k A(k)W^{jk}$$

for which FORT computes A(k), k = 0,1,...,N-1. From the A(k)'s one gets

$$\begin{aligned} A'(k) &= \tfrac{1}{2}\left(A(k) + \tilde{A}(N-k)\right) \\ A''(N-k) &= \frac{i}{2}\left(\tilde{A}(k) - A(N-k)\right)\,. \end{aligned} \tag{32}$$

The second equation is a modified form of (21'). Due to symmetry, the A'(k)'s and A''(k)'s need only be computed for half their indices.

It is useful to make a special case of the index k = 0 for which, as can be seen from above,

$$C(0) = \tfrac{1}{2}\left(\mathrm{Re}\,A(0) + \mathrm{Im}\,A(0)\right)$$

$$C(N) = \tfrac{1}{2}\left(\mathrm{Re}\,A(0) - \mathrm{Im}\,A(0)\right)\,.$$

Hence, the calculation in (32) is to be performed for $k = 1,2,\ldots,\frac{N}{2}-1$. To summarize the procedure:

Step 1. Call FORT with $M = \log_2 N$ to compute the Fourier transform A(k), of

$$X(j) = Y(2j) + iY(2j+1)\,, \qquad j = 0,1,\ldots,N-1.$$

Step 2. Let

$$C(0) = \tfrac{1}{2}\left(\mathrm{Re}\,A(0) + \mathrm{Im}\,A(0)\right)$$

$$C(N) = \tfrac{1}{2}\left(\mathrm{Re}\,A(0) - \mathrm{Im}\,A(0)\right)$$

Step 3. Do, for $k = 1,2,\ldots,\frac{N}{2}-1$

$$A'(k) = \tfrac{1}{2}\left(A(k) + \tilde{A}(N-k)\right)$$

$$A''(N-k) = \frac{i}{2}\left(\tilde{A}(k) - A(N-k)\right)$$

$$C(k) = \tfrac{1}{2}\left(A'(k) + \tilde{A}''(N-k)V^{-k}\right)$$

$$C(N-k) = \tfrac{1}{2}\left(\tilde{A}'(k) - A''(N-k)V^{k}\right) = \tfrac{1}{2}\left(\widetilde{A'(k) - \tilde{A}''(N-k)V^{-k}}\right)$$

Step 4. The complex C-vector contains the desired sin-cos coefficients as given in (31).

7. Computation of Real Fourier Series: To compute the real Fourier series (28) when given the a_k, b_k's, one does just the reverse

of the process described in the preceding section. If one reverses the order of the steps, solving for quantities on the right sides of the equations in terms of those on the left, one obtains the following procedure:

1. Let

$$C(0) = a_0/2$$

$$C(N) = a_N/2$$

$$C(k) = (a_k - ib_k)/2 , \qquad k = 1,2,\ldots,N-1.$$

2. Do, for $k = 1,\ldots,\frac{N}{2} - 1$,

$$A'(k) = C(k) + \tilde{C}(N-k)$$

$$A''(k) = (C(k) - \tilde{C}(N-k))V^k$$

$$A(k) = A'(k) + i\,A''(k)$$

$$A(N-k) = \tilde{A}'(k) + i\,\tilde{A}''(k)$$

3. $A(0) = (C(0) + C(N)) + i(C(0) - C(N))$

4. Call FORT with $M = \log_2 N$ to compute the Fourier series with coefficients A.

The resulting complex array will be

$$X(j) = Y(2j) + iY(2j+1) , \qquad j = 0,1,2,\ldots,N-1.$$

8. Convolutions and Correlations: Let X, A and Y, B be two transform pairs defined by

$$X(j) = \sum_k A(k)W^{jk}$$

$$Y(j) = \sum_k B(k)W^{jk} .$$

It is to be remembered here that all functions and their transforms are periodic of order N, so we may omit writing the range of indices. In calculations, it is suggested that values of one period, i.e., A(0), A(1),...,A(N-1) be stored in memory and, where a formula calls for

an index out of the range (0,N-1), one is to replace it by its value mod N. for example, if $0 \le k < N$, and a calculation calls for A(-k), then the program should refer to the index N-k, since A(-k) = A(N-k).

By substituting the Fourier series for X(j) and Y(j) we obtain

$$\rho^C(\ell) = \sum_j X(j)Y(\ell \pm j) = \sum_{j,k,k'} A(k')B(k)W^{jk'}W^{(\ell \pm j)k} .$$

Using the property of complex roots of unity,

$$\sum_j W^{j(k' \pm k)} = \delta(k', \mp k)\cdot N = \begin{cases} N & \text{if } k' = \mp k \\ 0 & \text{otherwise} , \end{cases}$$

one gets the convolution theorem,

$$\rho^C(\ell) = N\sum_k A(\mp k)B(k)W^{\ell k} . \qquad (33)$$

This says that the convolution of X and Y can be calculated by multiplying their Fourier transforms, A and B, and computing the Fourier series with their products as coefficients.

The convolutions given by these functions are, however, of periodic functions while in practice, one usually wants to treat the data as though it were zero outside the range (0,N-1) instead of being repeated. This difficulty is overcome by lengthening the arrays by appending L zeros to them where L is the maximum lag ℓ. Thus, to compute

$$\rho(\ell) = \sum_{j=0}^{N'-\ell} x(j)y(\ell+j)$$

where x(j), y(j), j = 0,1,...,N'-1 are given, we define

$$x(j) = 0 , \qquad y(j) = 0 , \qquad \text{for } N' \le j < N$$

where N = N'+L. Then the circular convolution, defined by (33), is our desired result. The ratio of the number of operations required by the direct method to the number required by the Fourier transform method is

$$\frac{(N-\frac{3}{2}L)(L+1)}{3N\log_2 N}.$$

If one wants all possible values of $\rho(\ell)$ then $L = N/2$ and this ratio is approximately

$$\frac{N}{12\log_2 N}.$$

Consider the case where $y(j)$ is a signal of indefinite length, and one wishes to compute the "filtered" signal

$$\bar{y}_\ell = \sum_{j=0}^{L} x_j y_{\ell-j} \qquad \text{for } \ell = 0,1,2,\ldots,\ldots$$

with filter characteristics determined by the weights x_j. Then, a practical procedure is to divide y into overlapping segments of length N and overlap L. Then x may be considered to be the filter to be applied to each segment. If the same filter is to be used with each segment, then N' can be chosen to minimize the number of operations. For this, we consider, in a rough approximation, the number of operations per filtered output, $2N\log_2 N/(N-L)$. These values are listed in Table 2.

From the table, one can see that if $L = 8$, the minimum number of operations, over all N, for the Fourier transform method is 13; so, the direct method is more efficient. At $L = 16$, the two methods are equally efficient, but for L greater than 16, the Fourier transform method is more efficient. For $L = 128$, the Fourier transform method requires only 23 operations per filtered output if one takes $N = 1024$ or $N' = N-L = 1024-128 = 896$. For larger L the speed ratio of the Fourier transform to the direct method rises rapidly. For $F = 1024$, it is about 33. Another characteristic brought out by this table is that the function $N\log N/(N-L)$ is so flat an N equal to a power of 2 can be selected to give practically the minimum number of operations.

For further details and procedures for computing convolutions and correlations via Fourier transforms, see references [7], [8], [9] and [14].

9. Other Applications: Several other applications, made feasible by the speed of the fast Fourier transform method, are in the solution of differential and integral equations.

R. W. Hockney [11] has demonstrated that, with a fast method of computing Fourier transforms he could solve a two-dimensional Poisson equation in one-tenth the time required by the best iterative methods.

W. T. Weeks [12] has published a method for computing an inverse Laplace transform as a series of Laguerre functions. O. Wing [13] has modified Week's procedure to one in which the Laguerre coefficients are obtained from a Fourier transform.

TABLE I - Calculated maximum and RMS error in computing the Fourier transform of a single complex exponential function $\exp(2\pi ijk/N)$, $j = 0,1,2,\ldots,N-1$, $k = 2$. The program was compiled by IBSYS, FORTRAN IV and run on the IBM 7094. The time for calculating one Fourier transform is given in minutes.

N	Errors in units of 10^{-8} Max. Error	Max. Error / $\log_2 N$	RMS Error	Time in minutes
512	4.47	.50	.0133	.008
1024	5.22	.52	.0072	.017
2048	5.22	.47	.0036	.037
4096	5.96	.50	.0019	.085
8192	6.71	.52	.0010	.175

TABLE II- Table of values of $2N\log_2 N/(N-L)$, the approximate number of complex multiply-adds required to compute each digital filter output for segments of length $N' = N - L$, where L is the number of filter weights. L is the number of operations per output by the direct method and is to be compared with table entries in evaluating the two methods.

L	N = 16	32	64	128	256	512	1024	2048	8192
8	16	13	14	15	17	18			
16		20	16	16	17	19			
24			19	17	18	19			
32				18	18	19			
64				21	21	21			
96					26	22	22	23	
128					32	24	23	24	
1024									30

10. References

1. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. of Comp., vol. 19, no. 90, (April 1965), pp. 297-301.

2. G. C. Danielson and C. Lanczos, "Some Improvements in Practical Fourier Analysis and Their Application to X-ray Scattering from Liquids," J. Franklin Inst., vol. 233, (April 1942), pp. 365-380 and pp. 435-452.

3. Phillip Rudnick, "Note on the Calculation of Fourier Series," Math. of Comp., vol. 20, no. 95, (July 1966) pp. 429-430.

4. HARM - "Harmonic Analysis Subroutine for IBM 7094, SDA no. 3425," SHARE Distribution Agency, Program Information Dept., IBM Corp., 40 Saw Mill River Road, Hawthorn, New York 10532.

5. F. Yates, "The Design and Analysis of Factorial Experiments," Harpenden: Imperial Bureau of Soil Science (1937).

6. I. J. Good, "The Interaction Algorithm and Practical Fourier Analysis," J. Roy. Statist. Soc., Ser. B., vol. 20 (1958), pp. 361-372; Addendum, v. 22, 1960, pp. 372-375, MR 21 no. 1674; MR 23 no. A4231.

7. T. G. Stockham, Jr., "High Speed Convolution and Correlation," Presented at the Spring Joint Computer Conference of the ACM, Boston, April, 1966.

8. G. Sande, "On an Alternative Method of Calculating Covariance Functions," unpublished Tech. Note, Princeton University, Princeton, N. J.

9. H. D. Helms, "Fast Fourier Transform Method of Computing Difference Equations Arising from Z-Transforms and Autoregressions," unpublished Tech. Note, Bell Telephone Laboratories, Inc., Whippany, New Jersey.

10. R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra, Dover Publications, New York, 1958.

11. R. W. Hockney, "A Fast Direct Solution of Poisson's Equation Using Fourier Analysis", J. ACM, 12 (January 1965) pp. 95-113.

12. W. T. Weeks, "Numerical Inversion of Laplace Transforms Using Laguerre Functions," J. ACM, 13 (July 1966), pp. 419-426.

13. O. Wing, "An Efficient Method of Numerical Inversion of Laplace Transforms", IBM Research Report no. NC 628.

14. J. W. Cooley, "Applications of the Fast Fourier Transform Method," Proceedings of the IBM Symposium on Digital Simulation, June 20-22, 1966.

15. W. M. Gentleman and G. Sande, "Fast Fourier Transforms--For Fun and Profit," Proceedings of the Fall Joint Computer Conference, San Francisco, Calif., Nov. 8-10, 1966.

Programming Information

DIMENSION Statements: The calling program should have the statement:

DIMENSION A(...), S(...)

If A is declared complex in a Type statement, then A should be given a dimension N, where N is the total number of complex elements in the largest array to be transformed. If A is not declared to be complex, one would have to specify its dimension as 2N.

The dimension of S is usually $\frac{1}{4}$N where N is the maximum number of points for all runs to be made on this job. However, FORT can use tables which are larger than necessary. In the section, USAGE, above, where it is shown how to double the capacity of FORT, or how to do real Fourier analysis, it is seen that one needs SIN and COS for twice as many angles (half the angular spacing) normally required by FORT. In this case, one can double the length of the S table by calling FORT with IFS = 0 and M in the arguments one larger than required by FORT for the transforms.

Arguments to FORT: To compute a Fourier transform, the calling program should have the statement

CALL FORT (A, M, S, IFS, IFERR)

A is the name of the array to be transformed. The elements of A are stored in the normal FORTRAN way of storing complex arrays, i.e., the real part of A(I), I = 0, 1, ..., N-1, is stored in the cell with index

2*I+1

and its imaginary part is stored in the cell immediately following.

M determines the length of the array, $N = 2^M$.

S is a vector containing $\text{SIN}(\frac{J\pi}{2.N})$, for J = 1, 2, 3, ..., N-1.

IFS is an integer telling FORT which of several functions to perform, as follows:

IFS = 0 to set up the SIN table only.
IFS = 1 to set up the SIN table and calculate the Fourier series.
IFS = -1 to set up SIN table and calculate a Fourier transform.
IFS = 2 to calculate a Fourier series only.
IFS = -2 to calculate a Fourier transform only.

IFERR is an error flag which is set equal to 1 by FORT if one of several errors listed below is detected in the argument M. It is set equal to 0 otherwise.

IFERR = 1 and IFS = 0, $\pm$1 means M is less than 1 or greater than 13.

IFERR = 1 and IFS = $\pm$ 2, means the SIN table is not large enough or has not been computed. At some point before calling FORT with IFS = $\pm$ 2, one must call it with an M greater than the present M.

IFERR = -1 if one is computing a SIN table unnecessarily.

Sample Program: The sample program provided reads in the parameters

M, K, NTR, IFPR

with a

FORMAT (4I3)

M is the argument M to FORT. K determines the test data X(j) = $\exp(2\pi ijK/N)$, j = 0, 1, 2, . . . , N-1. NTR tells how many transform pairs to perform. Setting NTR > 0 enables one to get better timing estimates. Setting IFPR $\neq$ 0 causes printing of test data and transforms.

The test program then computes the Fourier transform of the data and compares with the correct result

$A(k) = 1, \qquad A(k') = 0, \qquad \text{for } k' \neq k.$

The maximum difference, RMS difference, and time to set the SIN tables and compute a Fourier transform are printed. Then, if one wants better timing estimates, one sets NTR to a non-zero number. The program does NTR transform pairs, from A to X and X to A, and makes the same comparison with correct results. It prints the maximum and RMS difference and the time to do the NTR transform pairs, without the computation of SIN tables.

Card and Deck Format: The cards submitted were all punched on the 026 keypunch and the deck is set up for compilation on the IBM 7094 with the IBSYS operating system.

Deck Key

	No. of Cards
$EXECUTE	1
$IBJOB	1
TEST - SAMPLE PROGRAM, FORTRAN IV SOURCE DECK	
Identification - TEST 001 - TEST 093	93
FORT - FOURIER TRANSFORM, FORTRAN IV SOURCE DECK	
Identification - FORT 001 - FORT 205	205
$DATA and 7 input parameter cards	8
TOTAL NUMBER OF CARDS	308