DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF SOUTHERN CALIFORNIA

EQINFOS

(The Strong-Motion Earthquake Data Information System)

by

V.W. Lee and M.D. Trifunac

Report No. 82-01

A Report on Research Conducted Under a Grant from
the National Science Foundation

Los Angeles, California
April, 1982

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 General Introduction

The need for fast and efficient dissemination of strong- motion earthquake accelerograph data is supported by engineers, seismologists and geologists throughout the world. Analyses and interpretation of the accelerograph data have applications by earthquake engineers concerned with the development of earthquake resistant structural design, by geophysicists and seismologists dealing with the understanding of earthquake source mechanisms, and by public officials responsible for public safety. For example, a study made for the Office of Science and Technology of the Executive Office of the President of the United States of America concludes that:

"...efforts must be devoted for...the collection of records after destructive earthquakes, analyzing the motions to derive various engineering characteristics and making the original data and analyses available shortly after the earthquake. The fundamental aspect of all dynamic research in earthquake engineering is the ground motion to which the structure is subjected in a major earthquake..."

The objective of the present report is to present an earthquake data information system (EQINFOS, Fig. 1.1-1) which serves to:

(1) provide the data to the user community as quickly as possible,

(2) allow the outside users to have direct access to the data and computer programs through telephone, and

EQINFOS

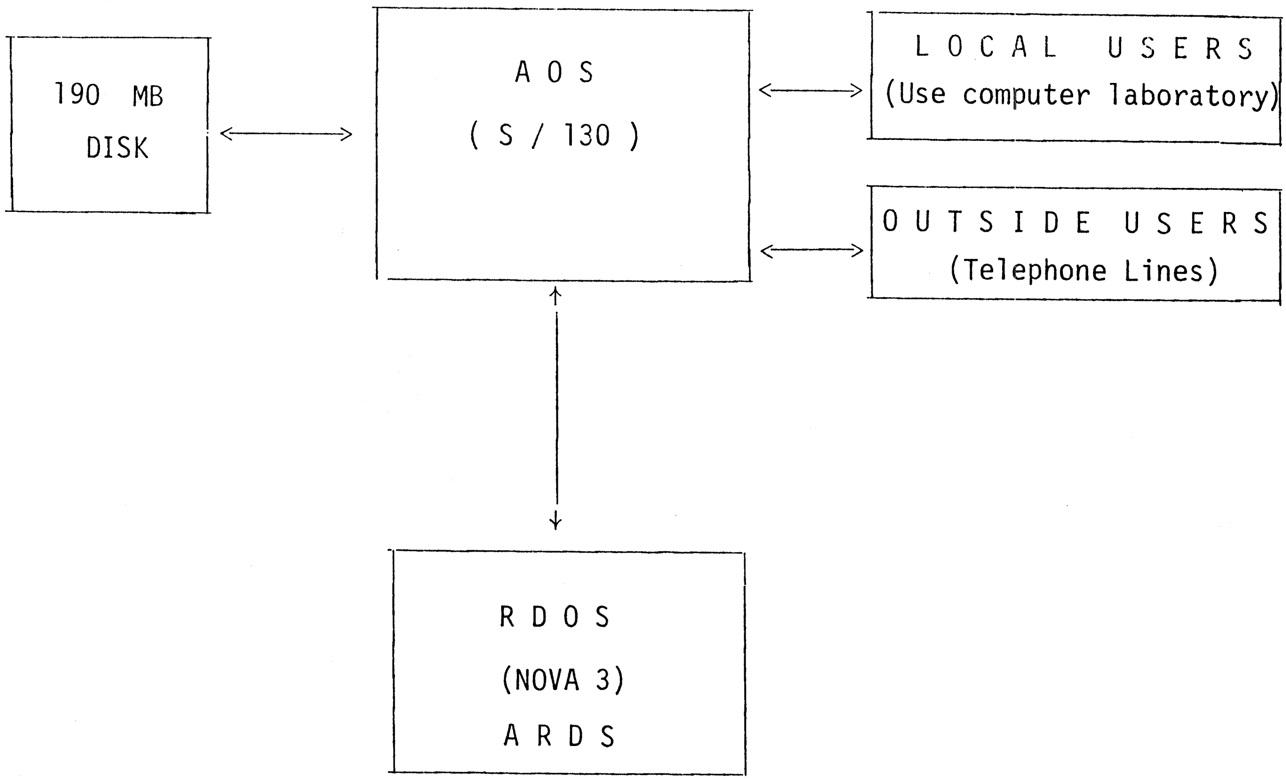


Fig. 1.1-1

(3) provide efficient means for data digitization (ARDS in Figure 1.1-1) and processing.

The organization of this system is shown in terms of a block diagram in Fig. 1.1-1. The system is run in a time sharing configuration with the Advanced Operating System (AOS) on the Data General Eclipse S/130 computer. The access to the system is either via direct terminals at USC or through standard telephone lines with the baud rates equal to 300. The files on strong motion accelerograms and the programs for data selection, processing and transmission are kept on to 190 MB disk.

Eclipse S/130 computer is hardwired to the Nova 3 computer operating on RDOS 10 MB disk system which has all software and hardware necessary for automatic strong motion accelerogram digitization - ARDS described by Trifunac and Lee (1979). This system provides the essential link between the new analog recordings on 70 mm and 7 inch film and the storage facility (on 190 MB disk of AOS) of the EQINFOS. Recently a number of improvements and new programs have been added to the ARDS system, and so, for completeness of this report, these are also outlined here and in the Appendix A. This information may be of interest to a user requiring digitization of an analog accelerogram on a film but can be ignored by those who only wish to learn how to access EQINFOS and receive data and information on what is in the system.

The main task of developing an automatic digitization system for strong-motion accelerograms (ARDS) is now completed. Under this system, sections of an accelerogram record on film can be reproduced onto a 10" by 10" negative, which is then put into the photo-densitometer for scanning. A scanning program "FILM", available on the NOVA 3 Data

General computer, is used to control the densitometer and to store the data read by it on disk. A program, "TRACE", is then used to read through the RAW DATA and to locate the coordinates of all the traces found. The user next employs the editing program, "TV", to read the traces, display them graphically on the Tektronix screen, and perform editing functions on the traces. Finally, the edited RAW DATA file of all acceleration traces is displayed on the plotter and transferred to a file on a magnetic tape for storage. This completes the automatic digitization of an earthquake strong-motion accelerogram record (Trifunac and Lee, 1979).

Via a direct line, the information in the NOVA 3 computer and all of its data files can be transferred into the AOS system on Eclipse S/130 and its 190 MB disk. Through telephone link the outside users can access this storage to retrieve or process selected strong motion data.

Some ten years ago, the digitization process available to Earthquake Engineers typically involved manual, hand digitization (Trifunac and Lee, 1973). Before digitizing, the film record or its enlargement were first placed on the digitizing table with the horizontal axis lined up by eye to an estimated zero axis on the digitizing table. Each trace was then digitized by placing the cross-hair manually on successive x-y coordinates of the trace. The digitizer converted the coordinates to digital values, which were directly punched on cards or recorded on tape. A set of computer programs were then used to read and plot the data to the same scale as that of the digitized record for direct visual check and comparison. Any errors found had to be corrected manually. Any portion that was digitized improperly had to be redigitized and replotted until the final plot agreed well with the digitized record. After this the RAW DATA was declared "ready" for routine computer processing (Trifunac and Lee, 1973).

Digitization of a record by the old manual method, and subsequent checking usually took about two to four days to complete. In contrast with ARDS, it now takes only two to three hours to digitize the record and to get to the stage ready for routine processing. Another important difference between the old manual and the new automatic digitization is the average number of digitized points per second. The old manual digitizing method would produce at most 30-50 digitized points per second, as compared to several hundred per second for the automatic digitization process. Thus the information on data of frequencies up to about 100 Hz can now be obtained.

The routine computer processing programs are available on the NOVA 3 and Eclipse S/130 computers at USC and can be used to process the raw digitized data directly. The typical Accelerogram Processing sequence consists of the following steps (Trifunac and Lee, 1979):

(A) Digitization of Raw Data from accelerogram records as described above.

(B) Volume I Processing. The timing marks are first checked for "evenness" of spacing and then smoothed by the 1/4, 1/2, 1/4 running average. The x coordinates of each trace are then scaled to units of time in seconds. Each fixed trace is next smoothed and subtracted from the corresponding acceleration trace, whose y coordinates are subsequently scaled to units of G/10.

(C) Volume II Processing. The scaled, uncorrected Vol. I accelerogram data are next corrected for instrument frequency response and base-line adjustment. The data are first low-pass filtered with an Ormsby filter having a cut-off frequency f_c = 25 Hz and a roll-off termination frequency f_r = 27 Hz. Instrument correction is next performed using the instrument constants ω and ζ determined from calibration tests for each acceleroraph transducer. The data are then baselined corrected by applying a high-pass Ormbsby filter. The cut-off and roll- off frequencies of the filter are usually determined from the signal-to-noise ratio of each acceleration component (Trifunac and Lee, 1979 I and II). The accelerogram data are then integrated twice to get the velocity and displacement. To avoid long period errors resulting from the uncertainties involved in estimating the initial values of velocity and displacement, the computed velocity and displacement data are high-pass filtered at each stage of integration, using the same Ormsby filter as that for baseline correction of the acceleration data.

(D) Volume III Processing. Using an approach based on the exact analytical solution of the Duhamel integral for successive linear segments of excitation, this final stage consists of calculating the Response and Fourier Spectra for 91 periods and 5 dampings from the Vol. II corrected accelerogram data. The times of maximum response for all periods and dampings are also recorded.

All the hardware components used in the automatic digitization system at U.S.C. are available commercially and can be assembled using different computers, provided the proper interface to the photodensitometer is designed. Although not the most advanced and efficient from the hardware viewpoint, this system does demonstrate the technical feasibility of the simple process of automatic digitization.

1.2 Notes on Recent Advances in the Automatic Digitization System

An efficient automatic digitization system is clearly essential when a large number of records have to be digitized especially after the occurrence of an earthquake of considerable magnitude.

The USC Data General computer system (Trifunac and Lee, 1979) has recently been upgraded to include an Eclipse S/130 computer with a 190 M-byte Disc Drive and operated by the Time-sharing Automatic Operating System (AOS). Concurrently, the California Division of Mines and Geology (CDMG) in Sacramento, under the California Strong Motion Instrumentation Program (CSMIP), has set up an Eclipse S/250 Data General Computer with a 190 M-byte Disc Drive and operated by the Real Time Disc Operating System (RDOS).

Since the development of the Automatic Routine Digitization System (ARDS) at U.S.C., numerous improvements in the execution and efficiency of the programs have been made. The program "TRACE", which is used to read through the RAW DATA created by "FILM" and to locate the coordinates of all the traces found, has been updated to have better techniques of identifying, for each x, the following set of pulses and for their joining with existing segments from previous x. The program "TV", which is to read the traces, display them graphically on Tektronix and perform editing functions on the traces, has been updated to have the additional function of checking all the "fillings" on the existing traces. This additional feature has been found to be particularly useful when the traces are made up of a lot of high frequency data. The program "SCRIBE", which originally plotted the digitized data and wrote the data onto magnetic tape, now has a new version, called "KSCRIBE", which also plots the digitized data, but writes the data onto a disk file directly. The new versions of "TRACE", "TV", and "KSCRIBE" are described in Appendix A.

The ARDS system at USC (Trifunac and Lee, 1979) was originally developed to digitize accelerograms recorded on 70 mm film. Under the ARDS system, up to four 10"-sections of the 70 mm record can be reproduced on a 10" by 10" negative film for one digitization run. Digitization of a film of four sections corresponds to digitizing a record of about 92 seconds in length, which is sufficient for practically all recordings.

Recently, many CR-1 systems have been deployed in buildings. This type of instrument records on a 7 inch film designed for recording acceleration components at different parts of a building under one central recording system. Almost half of the instruments used by CDMG are of this type, for example. One digitization of a 10" by 10" negative film by ARDS then corresponds to that of 10" section of a 7" record of about 25 seconds long. A need thus arose to upgrade the ARDS digitization system to take into accout multiple 10-inch sections of the CR-1 records. This updated Automatic Routine Digitization System is described in the Appendix B.

1.3 Outside Access to the System.

No routine outside access to a computer with detailed data and software system has been available so far. Yet it is advantageous for workers in the earthquake engineering community to have quick access to the data and to the related programs. So far, the typical way to acquire data and software has been through requests by mail. The data and/or software are usually loaded onto magnetic tape or cards and then mailed out to a user. Considerable bookkeeping, secretarial paper work and time of the computer operator are involved, making this a time-consuming and expensive activity.

Currently, any person in the United States or abroad who is interested on earthquake engineering strong-motion data can access the USC AOS computer system with simply an ordinary telephone line and an interactive keyboard terminal. Through the Earthquake Data Information System (EQINFOS), the outside user can request, for example, to have a specified set of earthquake data typed out on his terminal. The EQINFOS system allows the user to search for earthquakes of partiular time periods, locations and magnitudes, records at particular locations, epicentral distances, intensities and site conditions. Also, selection can be made for intruments with particular range of periods, dampings and sensitivities. Finally filtered data with particular band pass widths or with selected peak acceleration, velocity and displacement can be chosen.

Having obtained information and names of the data files which are of interest, the user can have the data files typed out on a terminal. Additional programs have also been developed to allow the user to have the Volume III spectral data plotted with line-printer format (line-printer plot) on paper or screen of the terminal.

Any user with a digital plotter interfaced with the terminal can also use the programs to plot the requested data. The data available for plotting include the corrected acceleration, velocity and displacement, and the Fourier and Response amplitude spectra. The programs so far developed for plotting through a remote terminal are designed to use the DP-1 or DP-11 COMPLOT Digital Plotters. These plotters are interfaced with the terminal through a HOUSTON PTC-5A or PTC-5B controller, as shown in Fig. 1.4-1.

User with a floppy-disk or cassette recorder interfaced with his terminal can also record the specific data from the remote terminal. To illustrate this, programs have been developed to allow the user to record a specified file of data onto a DATAMASTER II floppy disk recorder, for example. Through RS232C standard interface, the DMII recorder is interfaced between the data terminal and a modem (Fig. 1.4-2). The DMII represents a random access data storage which can provide flexibility to any RS232 compatible terminal. It uses a floppy disk which consists of 2,431 addressable records of up to 128 characters each, or a total of over 311,000 characters. With minor changes, the current programs can be modified to work with almost any RS232 compatible plotter or recorder.

Outside users can also carry out remote processing of their own data. Programs have been developed to read the input data that the user transmits from a DMII Recorder, or equivalent. Remote Data Processing software can then be used to do Volume II Data Correction and/or to

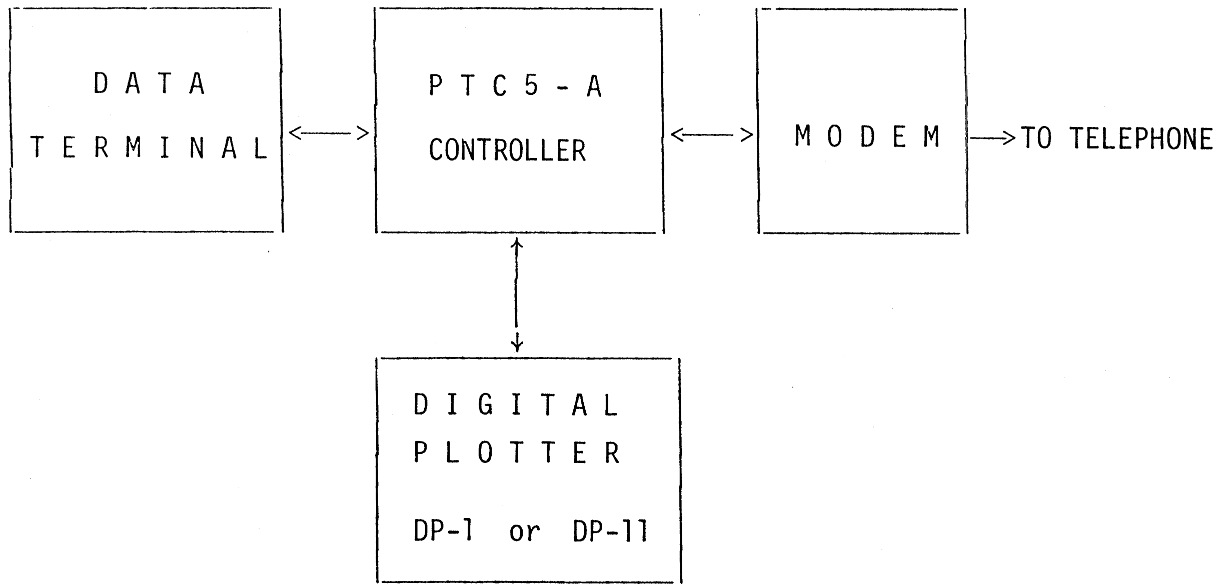


Figure 1.4-1

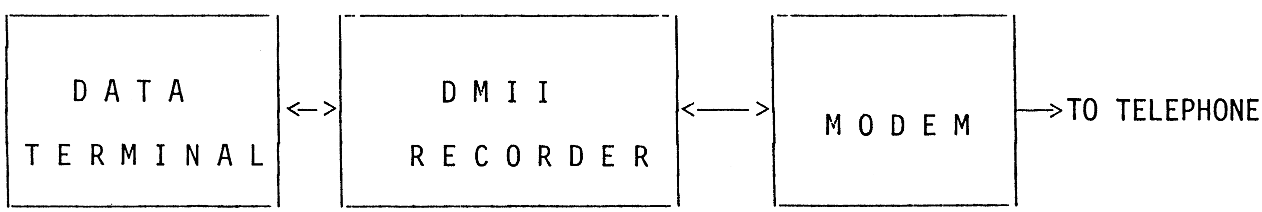


Figure 1.4-2

calculate Volume III Response and Fourier Spectrum amplitudes. Many users can thus afford to do data processing with hardware which with current prices costs less than about $10,000.00.

The EQINFOS as presened in this report has been designed mainly to serve small scale and real time users who have limited on-site computer capabilities and use small number of earthquake data files. Though it can be used for retrieval, processing and analysis of a larger data set, at present its 300 baud modem rate will represent a limitation for massive data transmission. For users in need of many strong motion accelerograms and with access to a computing center of their own, it continues to be more efficient to use magnetic tapes for data transmission.

We believe that the tasks which now can be performed by EQINFOS do represent almost all that is required for the elementary selection, retrieval, and remote input and processing of strong motion earthquake accelerograms. However, many features of the present system should first be tested by the critical users before further refinements, modification and extensions are considered. Since this system might best be tailored by the "average", "typical" user demands, the authors of this report are hopeful that those who choose to use the system will kindly contribute their critical remarks, comments and suggestions so that we can further improve the system to suit their needs. Thus if this report is viewed as a limited first step towards more modern and direct strong motion data dissemination system, we might consider our present task to have been accomplished.

CHAPTER II

THE STRONG-MOTION EARTHQUAKE DATA INFORMATION SYSTEM (EQINFOS)

2.1 User Account: Accessing the System

The system can be accessed by using an ordinary telephone and an interactive keyboard terminal that operates in full duplex mode and which uses ASCII character codes with the 300 baud rate. A user needs to know only how to dial the computer number and what to type to enter into the information system to begin using it. Once accessed, the system will offer a general introduction and will tell the user how to request more detailed information.

Perform the following steps to gain access to the system:

(1) Set the terminal to operate in the following modes:

a. transmission speed = 300 BUAD rate

b. full duplex

c. on line

(2) Telephone the USC AOS computer, (213)743-4623, and wait for a high pitched tone, then place the telephone handset in the cradle of the acoustic coupler. Watch for the "carrier" light to turn on, indicating that the terminal is properly receiving the signal.

(3) The computer will respond with the following line

```
****AOS REV 4.00/ TYPE NEW-LINE TO BEGIN LOGGING ON****
```

On typing the new-line key, the computer will respond with

```
AOS 4.00/EXEC 4.00 XX-MON-YR     XX:XX:XX    CON9
USERNAME:
```

To log on the user should communicate with the computer as follows:

```
USERNAME: EQINFO ↲
PASSWORD: EQUAKE ↲
```

where the process of pressing the return key is abbreviated by ↲. The password which the user typed into the terminal will normally not be echoed back onto the terminal. The computer will respond:

```
**********USC AOS SYSTEM DISK **********
```

The user is now logged onto the system*.

---

*The telephone number (213) 743-4623, the USERNAME:EQINFO and the PASSWORD:EQUAKE are active at the time of the writing of this report and will be maintained for a limited period of time to allow outside users to access the system, free of charge, test it and learn how to use it. Depending on the future use and organization of EQINFOS, the telephone number, the USERNAME and PASSWORD will change.

2.2 XEQ EQINFO

Information on the earthquake record data in the system will be available to users through the computer program "EQINFO". This program sorts earthquake and record data selected by each user. A sample run of the program in the next section will illustrate how it works in the example given. The program requests from a user information in three main categories:

(I) Earthquake Information: The program requests the user to specify desired bounds on earthquake parameters: time period, locations, magnitudes and maximum intensities.

(II) Information on Record Location: The program requests the user to specify pertinent parameters for records: locations, range of epicentral distances, intensities, and site conditions.

(III) Instrument and Record Data Information: The system requests the user to specify bounds on the parameters representing natural period, damping ratios and sensitivity of instrument transducers, digitized length, average number of points/sec., band-pass frequencies and widths, and times and magnitudes of peak acceleration, velocity and displacement of records chosen.

The request by the program for bounds on each parameter is of the form:

```
RANGE: (0)NO BOUND, (1) ≤, (2)WITHIN:[,],
       (3) ≥, OR (4)OUTSIDE:[,]
       INPUT IRANGE (0/1/2/3/4) [0]:
```

where 0 is the default value as indicated in [0], corresponding to no bound on the parameter in question.

An answer of 1 (≤) will result in the next question:

```
INPUT UPPER BOUND FOR "PARAMETER"
                                 ≤
```

where "PARAMETER" will be substituted by the parameter in question.

An answer of 2 (within[,]) will result in the next question:

```
INPUT THE BOUNDS FOR "PARAMETER"
                     WITHIN [,]:
```

An answer of 3 (≥) will result in the next question:

```
INPUT LOWER BOUND FOR "PARAMETER"
                                 ≥
```

Finally, an answer of 4 (OUTSIDE [,]) will result in the next question:

```
INPUT THE BOUNDS FOR "PARAMETER"
                     OUTSIDE [,]:
```

Having received the input bounds on all parameters, EQINFO will start searching for earthquakes and records with prameters within the bounds specified. The earthquakes and records found will be reported on the terminal to the user during execution. A complete dialogue on the bounds specified on each parameter and the earthquakes and records found will also be recorded on the file "EQINFOS.DT" available to the user at the end of the execution. Users with hardcopy line printer terminals can keep a copy of the file by typing on the terminal:

TYPE EQINFOS.DT ↲

The next section gives a listing of EQINFOS.DT for a sample run.

2.3 The Data File: EQINFOS.DT

This section presents a sample run of EQINFO. For illustration purposes, we select the following objective:

OBJECTIVE: Find all data from earthquakes dated before January 1, 1952 of magnitude ≥ 5.5 within the area of radius 500 km centered at 38° N and 118° W. Choose only those data from record locations with epicentral distances ≤ 1000 km. and M.M.I. within [5,8]. Include data only of the horiontal components and with peak acceleration greater than 100 cm/sec.

The user begins by typing

) EQINFO ↲,

answers each question and specifies the bounds as described in the previous section. The following pages represent a listing of the resulting data file:

EQINFOS.DT

which also gives a complete summary of the earthquakes and records found satisfying the above objective.

```
                 ******************************
                 *  EARTHQUAKE INFORMATION : *
                 ******************************
INFORMATION ON:
          DATES, MAGNITUDES, MAX. INTENSITIES OR LOCATIONS
OF EARTHQUAKES TO BE SPECIFIED? (Y/N) [Y]:


        DATES OF EARTHQUAKES:
RANGE: (0)NO BOUND, (1)BEFORE <=, (2)WITHIN [,], (3)AFTER >=,
 OR (4)OUTSIDE [,]? (0/1/2/3/4) [0]:
1

INPUT BOUND FOR DATES BEFORE:
            <= **,**,****  (MONTH,DAY,YEAR):

1  1 1952

 MAGNITUDES OF EARTHQUAKES:
RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
3
INPUT LOWER BOUND FOR MAGNITUDES
                             >=
     5.500

MAXIMUM INTENSITIES (M.M.I.) OF EARTHQUAKES:
RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
0

 LOCATIONS OF EATHQUAKES:
 WITH RESPECT TO A REFERENCE POINT LOCATION:
(0)NO BOUND, (1)SQUARE, OR (2)CIRCLE? (0/1/2) [0]:
2
INPUT LAT. & LONG. OF THE CENTER, ***.***N,***.***W :

38.000  118.000
INPUT LENGTH OF RADIUS IN KILOMETERS :
   500.00000

                 ************************************
                 *  RECORD LOCATION INFORMATION :  *
                 ************************************
INFORMATION ON:
          LOCATIONS, SITE CONDITIONS, ALLUVIAL DEPTHS, M.M.I.,
          OR EPICENTRAL DISTANCES
OF RECORD LOCATIONS TO BE SPECIFIED? (Y/N) [Y]:
Y

RECORD LOCATIONS:
WITH RESPECT TO A SPECIFIED REFERENCE POINT:
(0)NO BOUND, (1)SQUARE, OR (2)CIRCLE? (0/1/2) [0]:
0

SITE CONDITIONS OF LOCATION:
(0)SOFT ALLUVIUM, (1)INTERMEDIATE, (2)BASEMENT ROCK,
OR (3)NO RESTRICTION?    (0/1/2/3) [3]:
0

 DEPTH OF ALLUVIUM (FT):
RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
0

INTENSITIES (M.M.I.) OF LOCATION:
```

```
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
2
 INPUT BOUNDS FOR INTENSITIES (M.M.I.)
         WITHIN [****,****]:
     5.000     8.000

 EPICENTRAL DISTANCES OF LOCATION (KM):
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
1
 INPUT UPPER BOUND FOR EPICENTRAL DIST. (KM)
                           <=
  1000.000

            ******************************************
            *  RECORD & INSTRUMENT DATA INFORMATION:  *
            ******************************************
 INFORMATION ON:
        COMPONENTS, PERIODS, DAMPINGS, DIGITIZED LENGTHS &
        POINTS/SEC, RMS, CUT-OFF FREQUENCIES & WIDTHS, OR
        TIME & VALUES OF PEAK ACC., VEL. OR DISP.
 OF RECORDS TO BE SPECIFIED? (Y/N) [Y]:
Y

 COMPONENTS OF RECORDS:
 (0)HORIZONTAL & VERTICAL, (1)HORIZONTAL ONLY, OR (2)VERTICAL ONLY:
         (0/1/2) [0]:
 1
 INFO ON INSTRUEMENT PERIODS & DAMPINGS TO BE SPECIFIED? (Y/N) [Y]:
N
 INFO ON DIGITIZED LENGTHS, POINTS/SEC & RMS OF RECORDS
 TO BE SPECIFIED? (Y/N) [Y]:
N
 INFO ON CUT-OFF FREQUENCIES & WIDTHS OF RECORDS
 TO BE SPECIFIED? (Y/N) [Y]:
N
 INFO ON PEAK ACC., VEL. & DISP. OF RECORDS TO BE SPECIFIED? (Y/N) [Y]:
Y

 COMP'S PEAK ACCELERATION IN CM/SEC/SEC:
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
3
 INPUT LOWER BOUND FOR PEAK ACCELERATION (CM/SEC/SEC)
                           >=
   100.000

 TIME OF PEAK ACCELERATION IN SEC:
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
0

 COMP'S PEAK VELOCITY IN CM/SEC:
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
0

 TIME OF PEAK VELOCITY IN SEC:
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
0

 COMP'S PEAK DISPLACEMENT IN CM:
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
```

```
0

 TIME OF PEAK DISPLACEMENT IN SEC:
 RANGE: (0)NO BOUND, (1)<=, (2)WITHIN:[,], (3)>=, OR (4)OUTSIDE:[,]?
     INPUT IRANGE (0/1/2/3/4) [0]:
0




 EARTHQUAKES FOUND:
  # MON/DAY/YR TIME CODE  LATITUDE LONGTITUDE DEPTH MAG MMI    NAME

  1  3 10 1933 1754 PST 0 33 37  0 -117 58  0  16.0 6.3 9LONG BEACH, CALIF
 16  6 30 1941 2351 PST 0 34 22  0 -119 35  0  16.0 5.9 8SANTA BARBARA, CAL




 RECORD LOCATIONS FOUND:

REF.#
  EQ.#     LATITUDE  LONGTITUDE    AZIMUTH   EPICENTRAL DIST ALLUV
AB021        RECORD #    20
 0    1      34  0  0   -118 12  0     332 53 16      47.8    50.421171.0    6  21
AU299        RECORD #   146
 0   16      34 25 12   -119 12  0      80 23 59      35.9    39.315080.0    8 274
AV314        RECORD #   157
 0    1      34  3  0   -118 15  0     331  9 23      54.9    57.211380.0    7 289
AV315        RECORD #   158
 0    1      33 46 11   -118 11 35     308 39 52      27.2    31.6 9627.0    8 290








 DATA FILES FOUND:






 RECORD #     20      FILE NAME : V2Y3058.DT
CORRECTED ACCELEROGRAM  IIB021  33.001.0    COMP S08W   FILE 58 DERIVED FROM:

FILE   1  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-B, EERL 70-21
LONG BEACH EARTHQUAKE
MAR 10, 1933 - 1754 PST
IB021   33.001.0 T                                                    18
STATION NO. 288   34 00 00N,118 12 00W                                38
VERNON CMD BLDG                                                       15
COMP S08W                                                             9
LONG BEACH EARTHQUAKE   MAR 10, 1933 - 1754 PST                       47
EPICENTER  33 35 00N,117 59 00W                                       31
INSTR PERIOD = 0.0970 SEC  DAMPING = 1.000                            42
NO. OF POINTS = 1303  DURATION = 125.73 SEC                           43
UNITS ARE SEC AND G/10.                                               23
RMS ACCLN OF COMPLETE RECORD = 0.1016 G/10.                           43
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN    0.070  AND  25.000  CYC/SEC
  4947  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF    0.02   SEC.
PEAK ACCELERATION =    130.63277  CMS/SEC/SEC AT    1.8800 SEC
PEAK VELOCITY     =     29.04300  CMS/SEC     AT    6.6600 SEC
PEAK DISPLACEMENT =    -15.45642  CMS         AT    5.6000 SEC
```

```
INITIAL VELOCITY  =     9.85504  CMS/SEC       INITIAL DISP. =     3.76362  CMS

LONG BEACH EARTHQUAKE   MAR 10, 1933 - 1754 PST

IIB021  33.001.0   VERNON CMD BLDG   COMP S08W


 RECORD #    20     FILE NAME : V2Y3059.DT
CORRECTED ACCELEROGRAM  IIB021  33.001.0    COMP N82W   FILE 59 DERIVED FROM:

FILE   2  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-B, EERL 70-21
LONG BEACH EARTHQUAKE
MAR 10, 1933 - 1754 PST
IB021   33.001.0 T                                                     18
STATION NO. 288    34 00 00N,118 12 00W                                38
VERNON CMD BLDG                                                        15
COMP N82W                                                              9
LONG BEACH EARTHQUAKE   MAR 10, 1933 - 1754 PST                        47
EPICENTER  33 35 00N,117 59 00W                                        31
INSTR PERIOD = 0.1000 SEC  DAMPING = 0.592                             42
NO. OF POINTS = 1087  DURATION = 125.76 SEC                            43
UNITS ARE SEC AND G/10.                                                23
RMS ACCLN OF COMPLETE RECORD = 1.0134 G/10.                            43
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  4940  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF    0.02   SEC.
PEAK ACCELERATION =  -151.51968  CMS/SEC/SEC AT   2.1400 SEC
PEAK VELOCITY     =    17.31563  CMS/SEC     AT   1.9000 SEC
PEAK DISPLACEMENT =   -17.53442  CMS         AT   1.7000 SEC
INITIAL VELOCITY  =    -4.73019  CMS/SEC       INITIAL DISP. =     2.03554  CMS

LONG BEACH EARTHQUAKE   MAR 10, 1933 - 1754 PST

IIB021  33.001.0   VERNON CMD BLDG   COMP N82W


 RECORD #   146     FILE NAME : V2Y3436.DT
CORRECTED ACCELEROGRAM  IIU299  41.002.0   COMP N45E   FILE 436 DERIVED FROM:

FILE  16  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-U, EERL 73-26
SANTA BARBARA EARTHQUAKE
JUN 30, 1941  - 2351 PST
IU 299 -41.002.0  T                                                    20
STATION NO. 283     34 25 12N, 119 12 00W                              40
SANTA BARBARA COURT HOUSE, CALIFORNIA                                  37
COMP N45E                                                              9
SANTA BARBARA EARTHQUAKE    JUN 30, 1941  - 2351 PST                   52
EPICENTER -  34 20 00N, 119 35 00W                                     34
INSTR PERIOD = 0.100 SEC   DAMPING = 0.592    SENSITIVITY = 26.6 CM/G  69
NO. OF POINTS = 1971  DURATION = 61.820 SEC                            43
UNITS ARE SEC AND G/10                                                 22
RMS ACCLN. OF COMPLETE RECORD =  0.1892 G/10                           42
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  3082  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF   0.02   SEC.
PEAK ACCELERATION =    233.77721  CMS/SEC/SEC AT    0.2400 SEC
PEAK VELOCITY     =     21.77274  CMS/SEC     AT    0.3000 SEC
PEAK DISPLACEMENT =     -3.74226  CMS         AT    0.1800 SEC
```

```
INITIAL VELOCITY       -5.68851  CMS/SEC         INITIAL DISP. =     -2.35614  CMS

SANTA BARBARA EARTHQUAKE     JUN 30, 1941   - 2351 PST

IIU299  41.002.0   SANTA BARBARA COURT HOUSE, CALIFORNIA    COMP N45E






 RECORD #    146      FILE NAME : V2Y3437.DT
CORRECTED ACCELEROGRAM  IIU299  41.002.0   COMP S45E   FILE 437 DERIVED FROM:

FILE  17  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-U, EERL 73-26
SANTA BARBARA EARTHQUAKE
JUN 30, 1941  - 2351 PST
IU 299 -41.002.0  T                                                        20
STATION NO. 283      34 25 12N, 119 12 00W                                 40
SANTA BARBARA COURT HOUSE, CALIFORNIA                                      37
COMP S45E                                                                  9
SANTA BARBARA EARTHQUAKE     JUN 30, 1941   - 2351 PST                     52
EPICENTER -  34 20 00N, 119 35 00W                                         34
INSTR PERIOD = 0.100 SEC   DAMPING = 0.552    SENSITIVITY = 25.4 CM/G      69
NO. OF POINTS = 2402  DURATION = 61.817 SEC                                43
UNITS ARE SEC AND G/10                                                     22
RMS ACCLN. OF COMPLETE RECORD =  0.1641 G/10                               42
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  3091  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF   0.02   SEC.
PEAK ACCELERATION =   172.30933  CMS/SEC/SEC AT   0.3400 SEC
PEAK VELOCITY     =    21.69754  CMS/SEC      AT   0.4800 SEC
PEAK DISPLACEMENT =    -3.92315  CMS          AT   0.3200 SEC
INITIAL VELOCITY  =    -2.75970  CMS/SEC         INITIAL DISP. =     -1.36256  CMS

SANTA BARBARA EARTHQUAKE     JUN 30, 1941   - 2351 PST

IIU299  41.002.0   SANTA BARBARA COURT HOUSE, CALIFORNIA    COMP S45E






 RECORD #    158      FILE NAME : V2Y3472.DT
CORRECTED ACCELEROGRAM  IIV315  33.003.0   COMP SOUTH  FILE 472 DERIVED FROM:

FILE   4  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-V, EERL 73-27
LONG BEACH EARTHQUAKE
MAR 10, 1933  - 1754 PST
IV 315  33.003.0  P                                                        19
STATION NO.  131     33 46 11N, 118 11 35W                                 40
PUBLIC UTILITIES BLDG., LONG BEACH, CAL.                                   40
COMP SOUTH                                                                 10
LONG BEACH EARTHQUAKE    MAR 10, 1933   - 1754 PST                         48
EPICENTER -  33 35 00N, 117 59 00W                                         34
INSTR PERIOD = 0.105 SEC   DAMPING = 0.494    SENSITIVITY = 30.8 CM/G      69
NO. OF POINTS = 1971  DURATION =130.866 SEC                                43
UNITS ARE SEC AND G/10                                                     22
RMS ACCLN. OF COMPLETE RECORD =  0.1631 G/10                               42
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  4949  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF   0.02   SEC.
PEAK ACCELERATION =   192.73044  CMS/SEC/SEC AT   2.9800 SEC
PEAK VELOCITY     =   -29.42027  CMS/SEC      AT   6.7000 SEC
PEAK DISPLACEMENT =    22.72865  CMS          AT   5.7400 SEC
```

```
INITIAL VELOCITY  =    -1.95859  CMS/SEC        INITIAL DISP. =     -4.70624  CMS

LONG BEACH EARTHQUAKE   MAR 10, 1933  - 1754 PST

IIV315  33.003.0   PUBLIC UTILITIES BLDG., LONG BEACH, CAL.   COMP SOUTH





 RECORD #   158     FILE NAME : V2Y3473.DT
CORRECTED ACCELEROGRAM  IIV315  33.003.0   COMP WEST   FILE 473 DERIVED FROM:

FILE   5  OF UNCORRECTED ACCELEROGRAM DATA OF VOL. I-V, EERL 73-27
LONG BEACH EARTHQUAKE
MAR 10, 1933  - 1754 PST
IV 315  33.003.0  P                                                    19
STATION NO.  131     33 46 11N, 118 11 35W                             40
PUBLIC UTILITIES BLDG., LONG BEACH, CAL.                               40
COMP WEST                                                              9
LONG BEACH EARTHQUAKE   MAR 10, 1933  - 1754 PST                       48
EPICENTER -  33 35 00N, 117 59 00W                                     34
INSTR PERIOD = 0.105 SEC   DAMPING = 0.494    SENSITIVITY = 31.7 CM/G  69
NO. OF POINTS = 2021  DURATION =130.868 SEC                            43
UNITS ARE SEC AND G/10                                                 22
RMS ACCLN. OF COMPLETE RECORD =  0.1534 G/10                           42
ACCELEROGRAM IS BAND-PASS FILTERED BETWEEN   0.070  AND  25.000  CYC/SEC
  4949  INSTRUMENT AND BASELINE CORRECTED DATA
AT EQUALLY-SPACED INTERVALS OF   0.02   SEC.
PEAK ACCELERATION =  -155.99657  CMS/SEC/SEC AT   2.6800 SEC
PEAK VELOCITY     =    16.51239  CMS/SEC     AT   1.1800 SEC
PEAK DISPLACEMENT =    11.86521  CMS         AT   6.2800 SEC
INITIAL VELOCITY  =    -3.91311  CMS/SEC        INITIAL DISP. =     -0.84754  CMS

LONG BEACH EARTHQUAKE   MAR 10, 1933  - 1754 PST

IIV315  33.003.0   PUBLIC UTILITIES BLDG., LONG BEACH, CAL.   COMP WEST

1 # OF FILES FOUND :    6
```

CHAPTER III REMOTE DATA ACCESS

3.1 Remote Line-Printer Plot

The Strong-Motion Earthquake Data Information System (EQINFOS) described in the last chapter allows the user to obtain information on and the names of the data files requested. This chapter will describe ways available for the user to read, plot and record the data files from the remote terminal.

The simplest way to read a data file is to have the file typed out on the terminal. The command is

```
) TYPE XXXXXXX.DT ↲
```

where XXXXXXX is the name of the data file that the user specifies. User with a remote line-printer terminal will thus also be able to have the data printed on the paper using the same command.

Next, the user can have the spectral (Volume III) data plotted out on the terminal with line-printer format, using the program V3LPAPLT. The user can execute the program with the command

```
) XEQ V3LPAPLT ↲
```

and the computer will request the name of the spectral data file:

```
INPUT NAME OF SPECTRAL DATA FILE (5A2):
```

with user's response of

XXXXXXX.DT ↲

XXXXXXX.DT is the name of the data file here the user typed in to respond to the question. The computer will next ask the question:

(0) NARROW OR (1) WIDE MARGIN LINE-PRINTER PLOT? (0/1):

where narrow margin (0) corresponds to a horizontal x-axis length of 61 characters and wide margin (1) corresponds to a horizontal y-axis length of 91 characters. The user should use the wide margin plot only if his terminal has a margin wider than 100 characters to allow extra space for labelling. On answering 0 or 1 the computer will next respond with:

(0) FOURIER, (1) SPECTRAL RESPONSE OR (2) BOTH LINE PRINTER DATA
PLOT? (0/1/2):

where "0" will give only the Fourier spectrum plot, "1" will only give the Response Spectrum plot and "2" will give both the Fourier and Response Spectra plot. Having typed in the selection, the computer will next respond with the message:

ADJUST THE POSITION OF PAPER ON THE TERMINAL
PAUSE

which allows the user to start each plot on a fresh page. Having

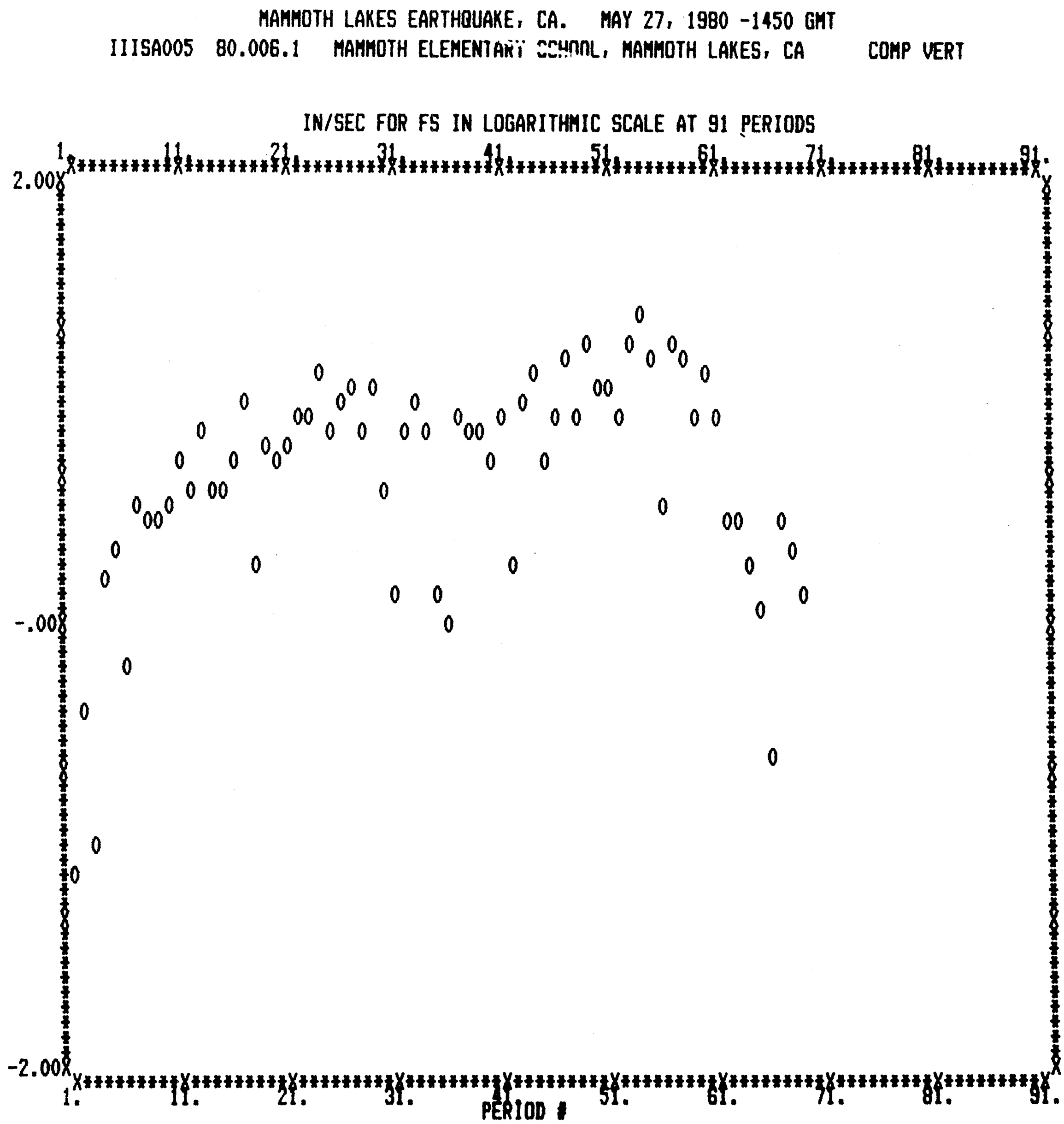

MAMMOTH LAKES EARTHQUAKE, CA.   MAY 27, 1980 -1450 GMT
IIISA005 80.006.1   MAMMOTH ELEMENTARY SCHOOL, MAMMOTH LAKES, CA   COMP VERT
IN/SEC FOR FS IN LOGARITHMIC SCALE AT 91 PERIODS
2.00
-.00
-2.00
1.
11.
21.
31.
41.
51.
61.
71.
81.
91.
PERIOD #

Fig. 3.1-1

MAMMOTH LAKES EARTHQUAKE, CA.   MAY 27, 1980 -1450 GMT

IIISA005 80.006.1   MAMMOTH ELEMENTARY SCHOOL, MAMMOTH LAKES, CA   COMP VERT

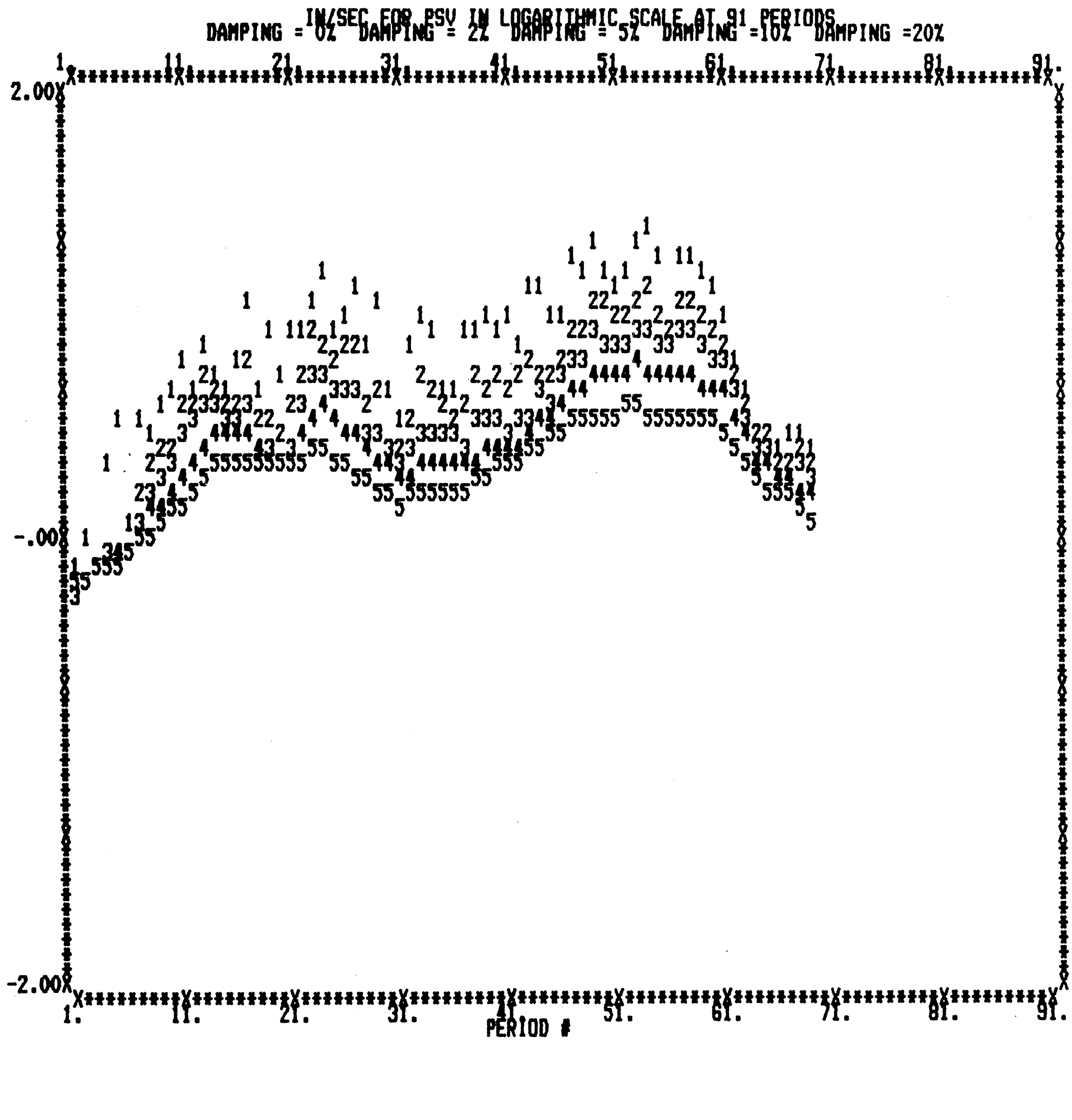


Fig. 3.1-2

adjusted the paper, the program will continue execution as soon as the user hits any key on the terminal. The program will repeat with the same message before initiating each plot until the last plot is completed. The following two line- printer plots in Figures 3.1-1 and 3.1-2 are examples of Fourier and Spectrum amplitude data plots of 1 component of Volume III data.

3.2 Remote Digital Plotting

In the past, the users needing plotting were often limited to expensive off-line plotting or to dedicated on-line plotters. These plotters were usually located in the computer room, inaccessible to many and thus often inconvenient to a user wishing instantaneous responses. With the advent of the time-sharing system, many users can now simultaneously access a computer from distant locations and thus can have remote plotting capabilities. Remote plotting has the following advantages:

(1) Economy - these plotters are less costly than large off-line systems.

(2) Accessibility - the plotters are now in the hands of the programmers who use them directly.

(3) Control - the plotters can be used without the help of an operator.

(4) Host Independence - the computer is free to do other tasks and is not held up by the plotter.

(5) Data Base sharing - many users in different geographical areas can share the same data.

Users equipped with simple remote plotting capabilities are now able to have a specified set of data plotted. The data available in EQINFOS for plotting include the Volume II corrected acceleration, velocity and displacement curves, and the Volume III Fourier and Response amplitude spectrum amplitudes.

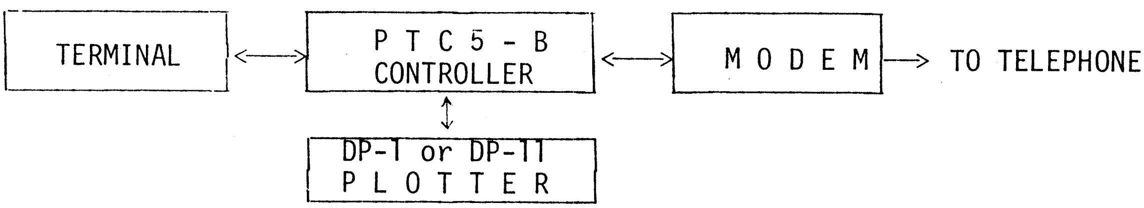
At present, many types of digital plotters are available on the market. The digital plotters are usually interfaced with the remote terminal via a controller that is either available as a separate device or is built into the digital plotter. The controller is a device that controls the position and movement of one or more pens on the digital plotter. It is usually a microprocessor that receives and sends ASCII coded information to and from an RS-232C interface. The codes used to generate plotter and pen commands are dependent on the model of controller used. In general, those codes correspods to the following commands

plotter initiation
pen up
pen down
new pen
pen movement by 1 unit of x-direction
pen movement by 1 unit of y-direction
symbol mode
exit

There are usually FORTRAN subroutines available for the HOST computer that are to be used to generate the above ASCII codes for the controller communication with the plotter. The programs developed for plotting Volume II an III data through a remote telephone access have been designed to use the DP-1 or DP-11 COMPLOT Digital plotter, which is interfaced with the modem through a Houston PTC-5A or PTC-5B controller. The subroutine package, COMPLT.LB, to generate the codes for the control of the plotter is available through Houston Instrument, Inc.

The following operations are required to set up the initial interface between the PTC-5B Controller, DP-11 Plotter and on RS-232C compatible remote terminal.

1) Place the PTC Controller between the Terminal and the Modem with proper RS-232C connecting cables, and connect the DP-11 Digital Plotter to the PTC Controller as shown in the following diagram.



2) On the Step Size Selection Switch of the plotter, select the step size of .005 in. Load the chart with blank plotting paper. The use of ball-point pen, supplied with the plotter, is recommended.

3) Perform the following operational check on the plotters as recommended by Houston Instrument, Inc.:

   a) Power ON unit, the power indicator mounted next to the switch should illuminate.

   b) Place the LOCAL/REMOTE switch in the LOCAL position.

   c) Place the PEN UP/PEN DN switch in th PEN DN position. The pen should lower and the indicator mounted next to the switch should illuminate.

d) Raise the pen with the PEN UP/PEN DN switch, the pen should lift.

e) By use of the LOCAL DIRECTION control move, both the pen carriage (Y) and the chart (X).

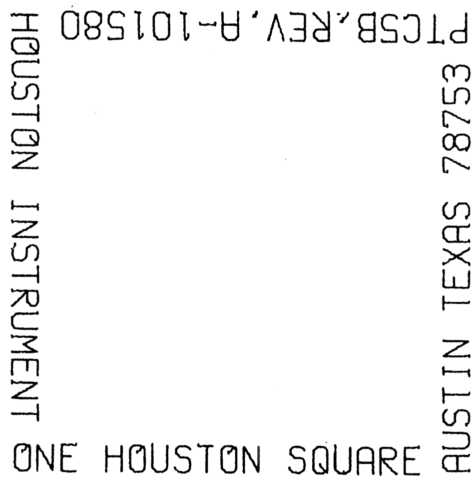
f) Place the PEN UP/PEN DN switch in the PEN DN position and repeat step e. Verify that pen trace is visible and even in both X and Y directions.

g) Move the CHART axis to a fresh sheet of paper and position the pen carriage to the desired plot start position by use of the LOCAL DIRECTION control.

h) Place the LOCAL/REMOTE switch in the REMOTE position, the indicator on the switch should illuminate.

The plotter is now ready for plotting.

4) On the front panel of the PTC Controller, turn the POWER ROCKER SWITCH ON. The power light in the upper left corner should illuminate. Turn the OPERATE/STANDBY rocker switch to STANDBY position. Perform a TEST PLOT as follows: turn the 3 position switch at the back, TEST/MUTE/ACTIVE to the TEST position and turn the front OPERATE/STANDBYswitch to operate. The plotter will respond with the following plot:



5) Turn the OPERATE/STANDBY switch back to STANDBY. At the back, put the TEST/MUTE/ACTIVE switch in the MUTE position, the BAUD RATE switch to 300. The PLOT SPEED is set by two 3 position slide switches at the back panel to the position that corresponds to 300 baud rate.

The interface is now set up and ready for plotting. Turn the OPERATE/STANDBY switch to OPERATE when everything is ready.

To plot Volume II corrected acceleration, velocity and displacement data of a specified data file, the user executes the following program:

```
) XEQ M2PLOT ↲
```

and the computer will request from the user the name of the data file:

```
INPUT NAME OF FILE TO BE PLOTTED (6A2):
```

with user response of

```
XXXXXXX.DT ↵
```

where XXXXXXX.DT is the name of a Volume II data file the user specifies. The computer will next type out the messages:

```
*****ON THE PTC CONTROLLER:
     SET STANDBY/OPERATE SWITCH ON FRONT PANEL TO OPERATE,
     ON BACK PANEL,
     SET TEST/MUTE/ACTIVE SLIDE SWITCH TO MUTE,
     SET BAUD RATE SWITCH TO 300, AND
     SET PLOTTER SPEED MATRIX SWITCHES TO 300;

*****ON DP HOUSTON PLOTTER:
     SET THE (INTERNAL) STEP SIZE SELECTION SWITCH TO
       A STEP SIZE OF .005 IN.,
     SET THE REMOTE/REMOTE ROCKER SWITCH TO REMOTE,
     MOVE THE CHART AXIS TO A FRESH SHEET OF PAPER, AND
     POSITION THE PEN CARRIAGE TO THE DESIRED START
       POSITION, THEN
     MOVE THE REMOTE/LOCAL ROCKER SWITCH TO REMOTE,
     AND FINALLY,
     WHEN ALL IS READY, HIT "RETURN" KEY ON YOUR TERMINAL.

PAUSE
```

The user is to check each of the steps as indicated and then hit the "RETURN" key on the Terminal. The program will then continue execution and plotting will start almost instantaneously. The plot will continue until the end of execution.

To plot Volume III Fourier and Response Spectrum Data file, the user executes the following program:

```
) XEQ M3PLOT ↵
```

and all subsequent steps and messages will be identical to those above. This completes the description of the remote digital plotting of one Volume II and one Volume III disk file.

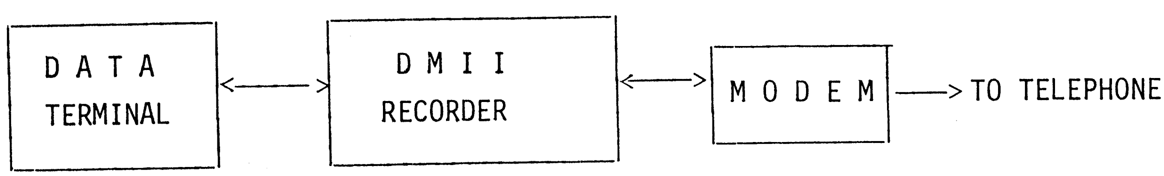
3.3 Remote Data Recording

In the past, those who wanted to receive data and the related sotftware were limited to first communicating by mail or telephone to make the first request. Subsequently, the needed data would be loaded on tapes or punched out onto cards and delivered to the user by mail. This may be a time consuming process for both the requesting and requested party and a lot of labor and time may be involved.

Users equipped with remote data recording capabilities will now be able to have a specified set of data files recorded immediately. The data files available at present include the Volume II corrected data and Volume III Spectrum amplitudes. The set of programs developed for EQINFOS is designed to allow the user to record data on a DATAMASTER II (DM II) floppy disk record (manufactured by the Western Telematic Inc.). Of course, other floppy disk devices can readily be used.

The following operations illustrate the steps to set up the interface between the DMII recorder and the terminal:

1) Interface the DMII recorder between the terminal and the modem with proper RS-232C connecting cables. Be sure that the correct BUAD rate (300) is selected on DMII.



2) Turn the Power switch of the Data Terminal ON, and put the

Data Terminal in the ON-LINE mode.

3) Turn the Power switch of the DATAMASTER II ON. The DMII is turned on by pressing the POWER ON button located in the upper left corner of the DMII.

4) Place all rocker switches on the right hand side of the DMII in the OFF or DOWN position.

5) With the disk drive door open, insert the flexible disk with the elongated cut-out end going in first and the envelope foldover down. Push the disk into the unit until it disengages the protective stop. Move the handle down until it "clicks" shut. The DISK READY light will go on.

6) Perform the following test from the Data Terminal keyboard to assure that connections between the Data Terminal and DMII are correct:

    Press the Control key on the Data Terminal and at the same time press the key CTRL-R, (R) followed by the key CTRL-T. This exercise should turn ON and OFF the RECORD light on the DMII.

With proper connections, the Data Terminal and DMII are now ready to interface with the USC AOS system computer. Put the DMII on STANDBY by pressing the "STANDBY" button located in the upper left corner of the DMII below the "POWER ON" button. Access the USC system as described in 2.1. Once the user is logged onto the system, he can record a specified data file by executing the following program:

```
) XEQ FLOPPY ↲
```

The computer will request from the user the name of the desired data file:

```
INPUT NAME OF FILE TO BE RECORDED (5A2):
```

with user response of

```
XXXXXXX.DT ↲
```

where XXXXXXX.DT is the name of the data file the user would like to record. The computer will next type out the messages:

```
***** ON DATA MASTER II: *****
      INSERT THE FLOPPY DISK INTO THE DISK DRIVE
      TURN "STANDBY" BUTTON OFF,
      PRESS "CLEAR", SELECT THE LINE NO., AND
      PRESS "RECORD", THEN
******  HIT "RETURN" ON DATA TERMINAL ******
```

PAUSE

The user is to check the four steps on DMII exactly as indicated on the messages and then hit the "RETURN" key on the DATA TERMINAL. The program will then continue execution with the specified file typed out on the terminal at the same time as it is being recorded onto the floppy disk. This will continue until the end of the file is reached. The user should then mark down the line number recorded on DMII. This completes the remote data recording of one disk file.

3.4 Remote Data Processing

Until recently, the data processing has been easily accessible only to large users with accounts in large memory computers. When the routine earthquake data procesing software was first developed, almost ten years ago (Trifunac and Lee, 1973), the programs were written for use in large computers like IBM 360 for example. The Data Processing software is now available on small mini-computers (Trifunac and Lee, 1979), with programs requiring less than 48K core size.

The users doing remote data processing will typically have uncorrected acceleration data of their own that requires Volume II type data correction and/or the calculation of Fourier and Response spectrum amplitudes. Two tasks thus have to be carried out: the first is to transfer the data to a disk file on the U.S.C. computer; and second, is to regenerate the data to be in the format as that of a routine Volume I uncorrected data file with appropriate heading information. The uncorrected data which a user may have are from an outside source and in a digitized form. We will assume here that this data is available on a floppy disk. The same Data Master II (DMII) recorder which was discussed in the preceeding section can now be used to play back the data onto a disk file interfaced with the Terminal the same way as described in the preceeding section (3.3). To transfer the data from the floppy disk onto a computer disk file, the user types:

) PLAYBACK ↲

The computer will request from the user the name of the disk file for the data:

```
INPUT NAME OF DISK FILE FOR THE DATA (5A2):
```

with user response of

```
XXXXXXX.DT
```

where XXXXXX.DT is the name of the disk file to be created. The computer will next type out the message:

```
***** ON DATA MASTER II:  *****
   INSERT THE FLOPPY DISK INTO THE DISK DRIVE,
   TURN "STANDBY" BUTTON OFF,
   PRESS "CLEAR", SELECT THE LINE NO.,
   HIT "RETURN" ON DATA TERMINAL,
   TYPE "COPY XXXXXXX.DT CONSOLE" ON  DATA TERMINAL
   HIT "RETURN" ON DATA TERMINAL, AND
   HIT "RUN" ON DATA MASTER II.
   THE FILE WILL BE TYPED OUT ON THE TERMINAL
   WHEN THIS FINISHES,
***** HIT CTRL-Q  CTRL-D ON DATA TERMINAL  *****

PAUSE
```

The user is to perform each of the steps as indicated. The data will then be loaded onto the disk file as it is being typed out on the terminal. When the end of file is reached, the user is to type CTRL-Q CTRL-D on the data terminal to close the file created.

The next step is for the user to create from the input disk file the Volume I uncorrected data file with appropriate format and appropriate heading information so that it is ready for Volume II data correction and Volume III Spectrum amplitude calculation. To do this, the user should have information on the data file he just created. This includes:

a) Earthquake Information, like earthquake name, epicentral coordinates, date and time, magnitude, etc.

b) Station Location Information, like station address and its coordinates; and

c) Station Instrument Information, like component directions, natural periods, dampings and sensitivities of each transducer.

With all this information, the user types onto the terminal:

```
) FIXVOL1
```

and the computer will request from the user the name of the input disk file:

```
INPUT NAME OF DISK FILE (5A2):
```

with user response of XXXXXXX.DT , the name of the file. Before reading the data, "F1XVOL1" will call subroutine "EQTITL" to request information on earthquake, station and record data, as in running of the Volume I data processing program. The reader should refer to the section on Volume I subroutine "EQTITL" in Trifunac and Lee (1979) for more detailed information. The heading, information will be written onto the Volume I data file "V1X01.DT".

The program will then request information on the input file. It is assumed that the data consists of time and acceleration coordinates which can be in any units, but will be scaled to units of seconds and G/10, as in a Volume I data file (Trifunac and Lee, 1979). The dialogue will be as follows:

```
     FOR T(I), X(I):  THE TIME AND ACCELERATION COORDINATES,
THE DATA ARE ASSUMED TO BE OF THE FORM
          (T(I), X(I), I=1,2,3...)  ON FILE,
INPUT TOTAL NO. OF PAIRS OF POINTS, NDATA:
INPUT FORMAT OF DATA PER LINE (5A2), LIKE (8F10.3):
INPUT NO. OF PAIRS OF POINTS PER LINE:
INPUT TIME SCALE IN SECONDS/TIME UNIT, TSCL:
INPUT ACCELERATION SCALE IN .1G/ACC UNIT,ASCL:
```

On completing the dialogue, the program will read the input file line by line, scale the data to units of seconds and G/10 and write the uncorrected data onto the file "V1X01.DT". This continues until the end of the file is reached.

The resulting Volume I data file "V1X01.DT" is now ready for Volume II processing. The user proceeds by typing

```
) RUNVOL2 ↲
```

which will execute the Volume II main program for data correction as described in Trifunac and Lee (1979). This will produce the Volume II corrected data file, called "IIX01.DT". This file is ready for Volume III Spectral data calculation. The user continues by typing

```
) RUNVOL3 ↲
```

which will execute the Volume III main program for Fourier and Response calculation as described in Trifunac and Lee, (1979). The output of Volume III calculations will be in the file IIIX01.DT. This completes the presentation of the remote data processing.

REFERENCES

Hudson, D.E., N.C. Nigam, and M.D. Trifunac (1969). Analysis of Strong Motion Accelerograph Records, Fourth World Conf. on Earthquake Engineering, Santiago, Chile.

Hudson, D.E., M.D. Trifunac, A.G. Brady and A. Vijayaraghavan (1971). Strong Motion Earthquake Accelerograms, II, Corrected Acceleration and Integrated Velocity and Displacement Curves, Earthquake Eng. Res. Lab., EERL 71-50, Calif. Inst. of Tech., Pasadena.

Hudson, D.E., M.D. Trifunac, and A.G. Brady (1972). Analyses of Strong Motion Earthquake Accelerograms, Response Spectra Vol. III, Part A, EERL 72-80, Calif. Inst. of Tech., Pasadena.

Trifunac, M.D. (1976). Preliminary Empirical Model for Scaling Fourier Amplitude Spectra of Strong Motion Accelerations in Terms of Earthquake Magnitude, Source to Station Distance and Recording Site Conditions, Bull. Seism. Soc. Amer., 66, 1343-1373.

Trifunac, M.D., and D.E. Hudson (1970). Laboratory Evaluation and Instrument Corrections of Strong Motion Accelerographs, Earthquake Eng. Res. Lab., EERL 70-04, Calif. Inst. of Tech., Pasadena.

Trifunac, M.D., F.E. Udwadia and A.G. Brady (1971). High Frequency Errors and Instrument Corrections of Strong Motion Accelerograms, Earthquake Eng. Res. Lab., EERL 71-05, Calif. Inst. of Tech., Pasadena.

Trifunac, M.D., F.E. Udwadia, and A.G. Brady (1973). Analysis of Errors in Digitized Strong Motion Accelerograms, Bull. Seism. Soc. Amer., 63, 157-187.

Trifunac, M.D., and V.W. Lee (1973). Routine Computer Processing of Strong Motion Accelerograms, Earthquake Eng. Res. Lab., EERL 73-03, Calif. Inst. of Tech., Pasadena.

Trifunac, M.D., and V.W. Lee (1974). A Note on the Accuracy of Computed Ground Displacements from Strong Motion Accelerograms, Bull. Seism. Soc. Amer., 64, 1209-1219.

Trifunac, M.D., and V.W. Lee (1978). Uniformly Processed Strong Earthquake Ground Accelerations in the Western United States of America for the Period 1933 to 1971: Corrected Acceleration, Velocity and Displacement Curves, Dept. of Civil Engineering, Report No. E 78-01, USC, Los Angeles.

Trifunac, M.D., and V.W. Lee (1979). Automatic Digitization and Processing of Strong Motion Accelerograms, Dept. of Civil

Engineering, Report No. 79-15 I and II, University of Southern California, Los Angeles.

APPENDIX A

THE UPDATED AUTOMATIC ROUTINE DIGITIZATION SYSTEM (ARDS) I

This appendix describes the recent improvements in the execution and efficiency of the various programs since the development of the Automatic Routine Digitization System (ARDS) at U.S.C. (Trifunac and Lee, 1979). These improvements have been motivated and then developed through the actual use of the ARDS in digitization of several recent records from the Imperial Valley Earthquake of 1979 and Livermore Earthquake of 1980.

A.1 NTRACE

When the first stage of digitization, FILM, is completed, the program, TRACE, is used to read the raw data file, "SCANS", created by FILM, to locate the coordinates of all the traces found, and to write the center-line amplitudes of the traces onto disk files.

The main function of TRACE (Trifunac and Lee, 1979), at each x is to deal with, the following pulse (and its subpulses) and to attempt to associate those with segments from previous x already in existence. The pulse is sent to the subroutine "BUILD", which does a preliminary check. First subroutine TOUCH is called to determine if the pulse overlaps with the width of the last pulse of any existing segment. A check is then made by subroutine REFCHK to see if the center of the pulse found matches closely with the end of any existing reference line segment. The pulse is immediately associated with the reference line segment if these two criteria are met. If this fails the "reference-line" test, but overlaps the end pulse of some other segment, with the pulse's center lying in the "expected direction" the given segment is "pointing", and if there is no overlap with any other segment, the pulse is associated with the segment. If, however, more than one segment overlaps the pulse, the pulse is examined in greater detail. First subroutine PFORM is called to examine the pulse for multiple subpulses. Each subpulse is then examined for matching with ends of existing segments as above. The combination with the highest "score" wins, and that subpulse will be associated with the matching segment. When all these tests fail, a new segment is created from the pulse.

The old version of TRACE may fail to associate the pulse with segments when the frequency of the data is high. NTRACE deals with the pulse in exactly the same way as above, but when all the above tests fail, it undergoes one more test to try to match the pulse with any of the existing segments before quiting to create a new segment from the pulse. Subroutine REFCHK is called again, and this time the pulse is tested to see if it is within a specified rectangular area of the ends of any of the existing segments. The size of the rectangular area specified starts out as 3 densitometer units in x and 15 densitometer units in y (Dx=3., Dy=15.) If this fails, the height is doubled (DY=30.) and the test is repeated. The closest segment within the rectangular area will be associated with the pulse. The test is repeated until a maximum of 60 units in height (DY=60.) is reached. This new test will serve to associate pulses with existing segments for high frequency impulsive acceleration traces.

The following pages represent a listing of the updated subroutine REFCHK, where the above new test is performed.

A.2 NTV

TV is the next software stage of ARDS that follows TRACE. It reads the traces, displays them graphically on Tektronix terminal and provides the editing capabilities necessary to check and/or repair data misinterpreted by TRACE or to fill in the missing data.

A typical editing session (Trifunac and Lee, 1979) involves examining the segments in one of the film-strip zones of the scan from margin to margin. Using the cross-hairs, any trouble spot may be blown-up at greater resolution. By displaying the raw data in areas of confusion, the operator can edit existing segments to correct for any misinterpretation by TRACE. TV is now updated to NTV, which has the additional function of checking all the "fillins" on the existing traces. This additional feature has been found to be particularly useful when the traces are made up of a lot of high frequency data. In this new version of TV, one edits the traces as before.
When the question appears:

```
DISPLAY "A"LL SEGMENTS,
     S"OME, or "N"ONE.
```

A response of "A" or "N" will proceed as before. A response of "S" will be followed by an additional question:

```
CHECK FILLIN? (Y/N):
```

An answer of "Y" (YES) will result in the segment to be displayed to have all "fillins" checked. This will be described below. The next question is to request the segment numbers.

SEG:

up to 5 segment numbers can be input. However, if the user has requested "CHECK FILLIN", it is recommended that only one segment number be input.

Assume that the user has requested "CHECK FILLIN", and input a segment number. NTV will proceed to display the segment and as it comes to a point where there is "FILLIN", the program will automtically blow up that area, display all the segments there and the RAW DATA in the area. For example: see Figure A2-1,

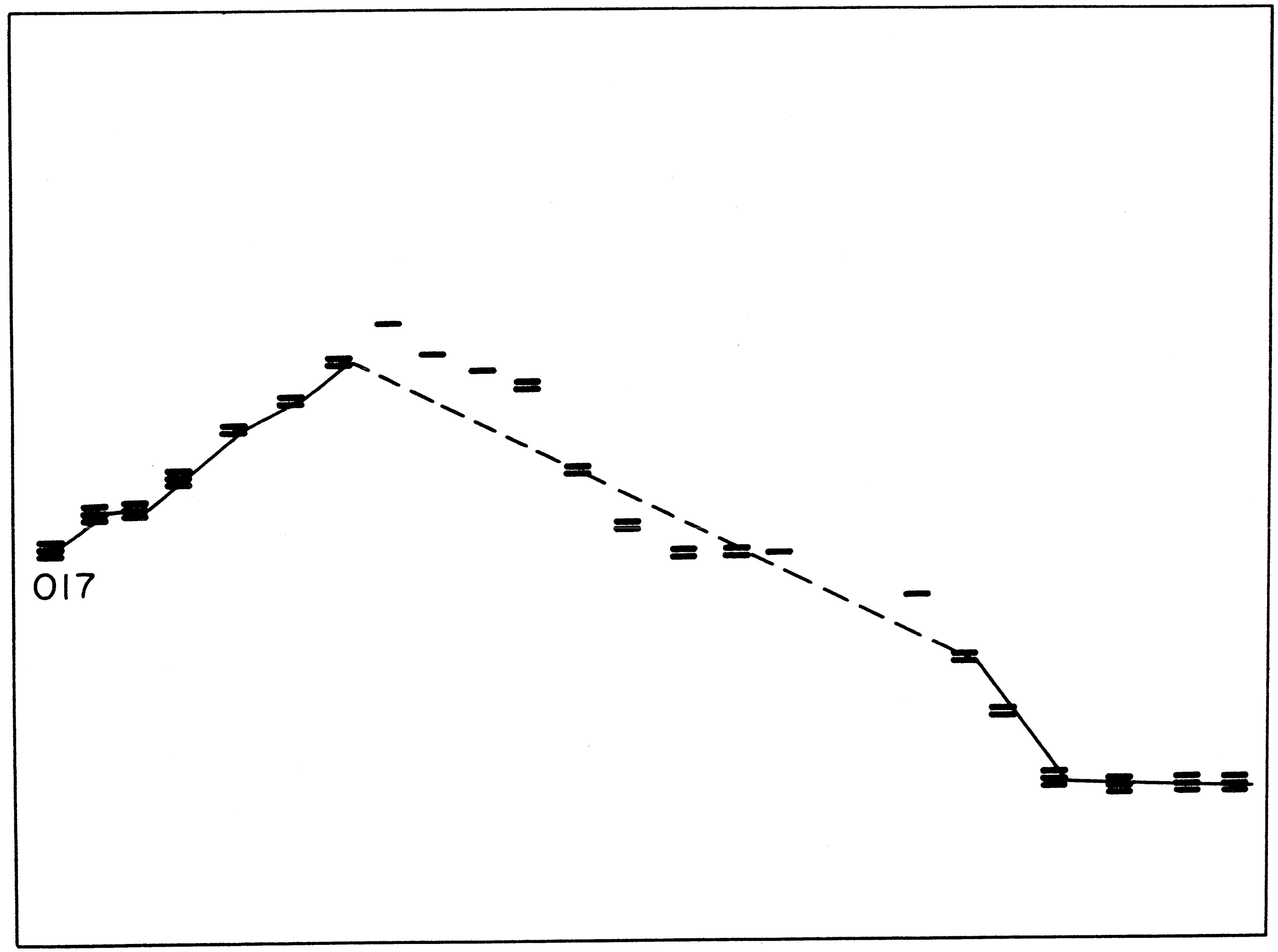


Figure A2-1

and a question will appear:

```
"4"-STEPS MENU,
"A"UTO-DIGITIZATION,
"B"LOW-UP, "R"AW DATA,
"P"ROCEDURES, "S"CALE, "Q"UIT:
```

the new command, "4"-STEPS MENU, is designed specially to take care of inappropriate "FILLINS", as is the case for the above picture. The 4 steps involve automatically (1) splitting both ends of the portion of the segment where the "FILLINS" are, (2) deleting the "FILLINS", (3) automatically digitizating the missing portion, and (4) merging the new portion with the existing segment. Suppose the user types in "4" to request for the "4"-STEPS MENU, TV will respond with the question:

```
SEG: 017 ↵
```

to which the user types in the segment number 017. Cross-hairs will then be activated by TV, expecting the user to input the coordinates of three corners of a rectangule where the "FILLINS" portion will be deleted and reigitized. For example, in the above picture, user can define the window as in Figure A2-2.

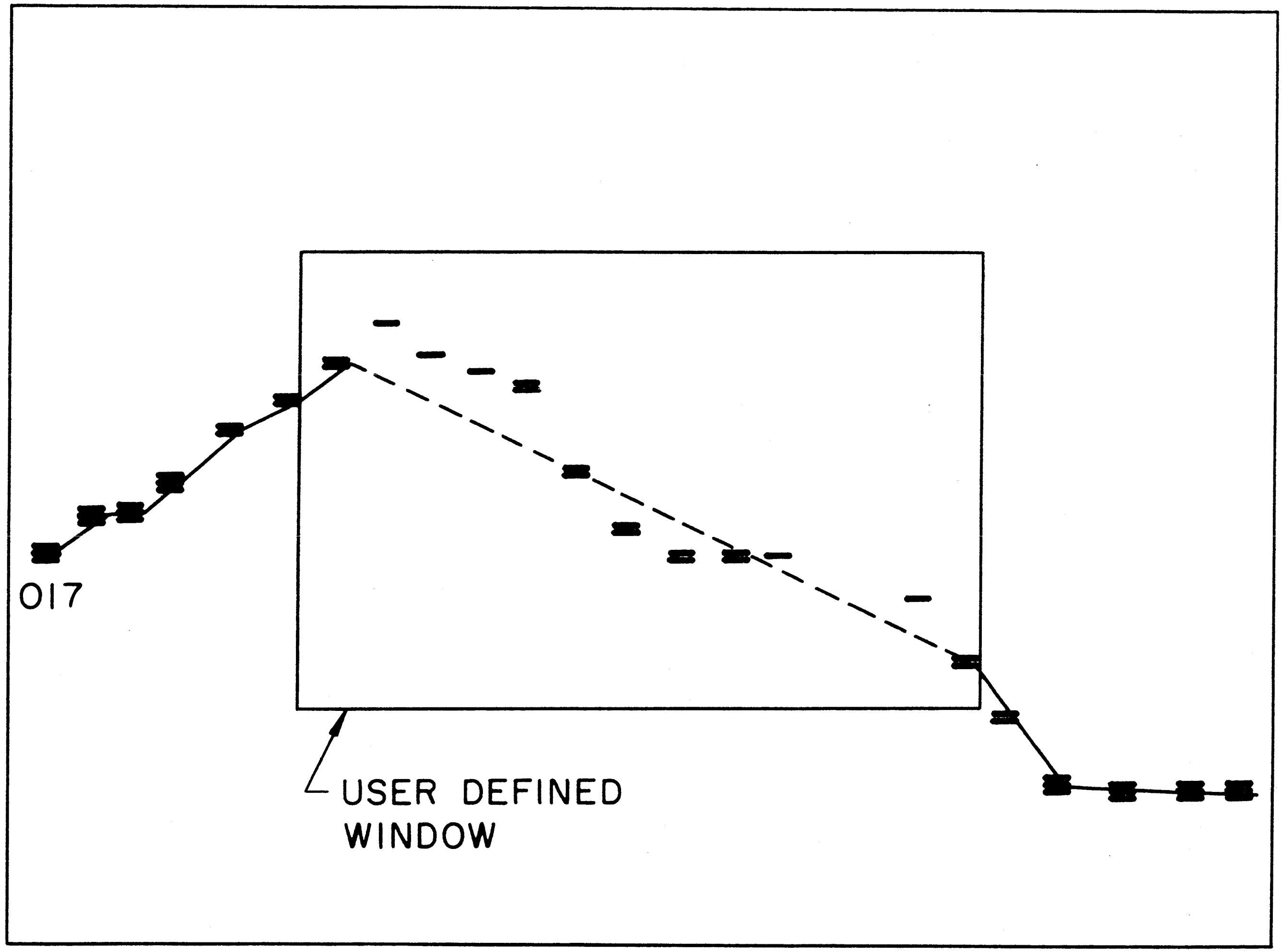


Figure A2-2

TV will then automatically perform the 4 steps indicated above. In splitting both ends of the "FILLINS" portion, intermediate segments will be created, and before deleting the middle "FILLIN" portion, the user is asked to verify with:

```
CONFIRM (Y/N):
```

Having responded, NTV will finish the "4-steps menu" and on completion, the segment 017 will then be a solid segment where it was a dashed one before (Figure A2-3). The same question will appear:

```
"4"-STEPS MENU,
"S"UT-DIGITIZATION,
"B"LOW-UP, "R"AW DATA,
"P"ROCEDURES, "S"CALE, "Q"UIT:
```

The user will now hit the "RETURN" (or "NEW LINE") key, and the same segment will now continue to be displayed, passing the first "FILLIN" portion until the next "FILLIN" portion is encountered.

The above process will be repeated. This continues until all the "FILLINS" are checked or the end of the segment is reached. "CHECK FILLINS" is a fairly time-consuming and involved process, and the user is adviced to first pick a prticular portion of the segment by "B"LOW-UP before requesting for "C"HECK-FILLIN.

The following pages present a listing of the updated main program and subroutines:

NTV.FR
NGETSEG.FR

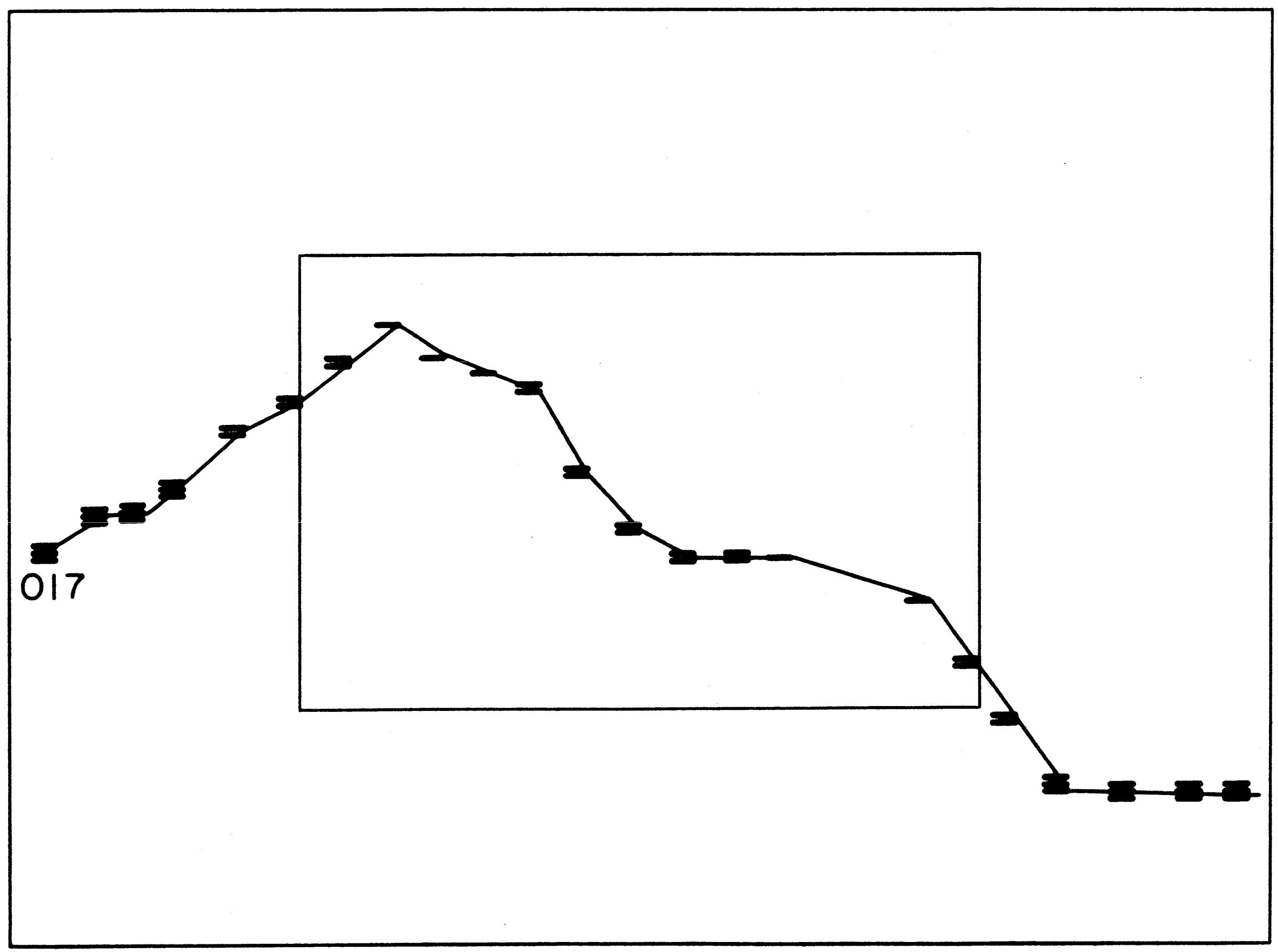


Figure A2-3

NBOUND.FR
NCHKSEG.FR
NMENU.FR
S4EDIT.FR

```
C       TITLE: TV.FR
C
C  TV IS A GRAPHICAL EDITOR.  IT ALLOWS MANIPULATION OF THE
C  SEGMENTS OF SEISMIC TRACES IDENTIFIED BY THE FILTERING
C  PROGRAM, "TRACE".
C
C  TV REQUIRES TWO ASCII FILES, "SUMUP", CREATED BY TRACE,
C  AND "REFS", CREATED BY "FILM", THE FILM-SCANNING ROUTINES
C  THAT USE THE OPTRONICS DENSITOMETER.
C
C  TV ALSO REQUIRES THE "SG--" FILES CONTAINING THE Y-COORDS
C  OF THE SEGMENTS, AND "PARAM", THE SEGMENT PARAMETER
C  FILE.  THESE ARE CREATED BY TRACE.
C
C  TV WRITES THREE FILES FOR ITS OWN USE AND TC BE PASSED
C  ON TO THE TAPE OUTPUT/HARDCOPY PLOTING
C  ROUTINES:  "MP--", THE MAP FILE, "SG--", "TPFD--", THE
C  DATA FILE CONTAINING FIDUCIAL COORDINATES PASSED ON FROM
C  THE FILM-SCAN, AND THE SEGMENT SERIAL NUMBERS TO BE FOUND
C  USED IN THE MAP FILE AND WHICH, IN TPFD, ARE ASSOCIATED
C  WITH THE ZONE--FILM STRIP--IN WHICH THEY ARE LOCATED.
C
C  TV CALLS PINIT, REFER, MAPMAK, ZONER, MAGFY, MENU, BOUND,
C  RAWDAT, MANIPU, AND ADIEU.
C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C

      PARAMETER SCREEN=1.31154       ; 1023/780: SCOPE WIDTH/HEIGHT
      PARAMETER SCREO2=.65577               ; SCREEN/2
      PARAMETER SCREI2=.38123               ; (780/1023)/2

        COMMON/LOCAL/BX(4),BY(4),XPAIR(2),YPAIR(2),XLAST,RPERX
      COMMON/BUFFS/PARAM(20),SEGPTS(64)
      COMMON/LOS/  NSEGS,NSER(500),NLIST,LIST(5),NSWER
      COMMON/MSGS/ IVER,IHOR,IX,IY
      COMMON/VERS/MSG(15)
      DIMENSION KARS(72),BXS(4),BYS(4)

      DATA MSG(1),MSG(2),MSG(3),MSG(4),MSG(5),MSG(6),
     1      MSG(7),MSG(8),MSG(9),MSG(10),MSG(11),MSG(12),MSG(13),
     2      MSG(14),MSG(15)/
     1      86,69,82,83,73,79,78,32,49,46,49,32,32,32,32/


200      CALL PINIT

      CALL OPEN(ICH,"SUMUP",1,IER)     ;READ DATA FROM FILTER PROGRAM
      READ(ICH,1000) XL,XR,YB,YT,NSEGS,NRECS
1000     FORMAT(4F10.0,/,I3,I6)
      CALL CLOSE(ICH,IER)

      SMARGIN=((XR-XL)/1023.)*10.     ;CALC A 10-SCREEN-UNIT MARGIN
      XR=XR+SMARGIN
      XL=XL-SMARGIN
      YT=YT+SMARGIN
      YB=YB-SMARGIN

      XWIDTH=XR-XL
      YWIDTH=YT-YB
      XMIDL=(XL+XR)/2.
      YMIDL=(YB+YT)/2.
      RPERX=FLOAT(NRECS)/XWIDTH       ; RECS PER INCREMENT OF X
      IF(RPERX.LT.1.)RPERX=1.

      CALL INITT(0)
      CALL TERM(2,1024)
```

```
      CALL SETBUF(4)

      CALL CHRSIZ(1)
      CALL MOVABS(350,400)
      CALL ANMODE
      CALL ADEOUT(12,MSG)       ; DISPLAY VERSION NO.
      CALL ADEIN(NCHAR,KARS)       ; BEGIN WHEN A CHAR ENTERED
      CALL CHRSIZ(4)
      IF(NCHAR.EQ.0)GO TO 10
      IF(KARS(NCHAR).EQ.81)STOP       ; IF A "Q", QUIT

10      IVER=LINHGT(1)
      IHOR=LINWDT(1)

      CALL REFER               ; READ REFERENCE FILE
      CALL MAPMAK               ; CREATE THE MAP
      CALL ZONER               ; ASSIGN ZONES TO SEGMENTS

            ; SET THE PRIMARY WINDOW SO THE SQUARE GRID IS
            ; NOT DISTORTED.  (ALL WINDOWS SET BY THE
            ; OPERATOR WITH CROSSHAIRS WILL BE DISTORTED
            ; TO FILL THE RECTANGULAR SCREEN.)

100     BX(1)=XL
      BX(2)=XL
      BX(3)=XR
      BX(4)=XR
      BY(1)=YT
      BY(2)=YB
      BY(3)=YT
      BY(4)=YB

      IF(XWIDTH.GT.(SCREEN*YWIDTH))GO TO 101
      IF(XWIDTH.LT.(SCREEN*YWIDTH))GO TO 102
      GO TO 103

101     BY(1)=YMIDL + (XWIDTH*SCREI2)
      BY(2)=YMIDL - (XWIDTH*SCREI2)
      BY(3)=BY(1)
      BY(4)=BY(2)
      GO TO 103

102     BX(1)=XMIDL - (YWIDTH*SCREO2)
      BX(3)=XMIDL + (YWIDTH*SCREO2)
      BX(2)=BX(1)
      BX(4)=BX(3)
      GO TO 103

103     CALL MAGFY(BX,BY)       ; SET SCREEN TO CURRENT WINDOW

150     NSWER=0
      CALL MENU(0,IANS,IZONE)       ;GET SEGS TO BE DISPLAYED
      IF(IANS.EQ."Q")GO TO 300
      IF(IANS.EQ."E")GO TO 150
        IF(NSWER.EQ.89)GO TO 151
      IF(IANS.EQ."A")CALL GETSEG(IZONE,NSEGS,NSER)
      IF(IANS.EQ."S")CALL GETSEG(IZONE,NLIST,LIST)
        GO TO 155
151     CALL CHKSEG(IZONE,NLIST,LIST)
155     CONTINUE

20      DO 21 J=1,4
      BXS(J)=BX(J)
      BYS(J)=BY(J)
21      CONTINUE
      CALL BOUND(BX,BY,JOB)               ; FIND OUT WHAT TO DO
```

```
      IF(JOB.EQ.0)GO TO 100

      GO TO (1,2,3,4,5,6),JOB

1       CALL RAWDAT(BX,BY,RPERX)       ; DISPLAY RAW DATA
30        DO 31 J=1,4
      BX(J)=BXS(J)
      BY(J)=BYS(J)
31        CONTINUE
      GO TO 20

2       CALL MAGFY(BX,BY)
C       CALL GETSEG(7,NSEGS,NSER)
C       GO TO 20
          GO TO 150
3       CALL AUTODG(BX,BY,RPERX)      ; AUTO-DIGITIZATION
        GO TO 30
4       CALL MANIPU              ; MANIPULATE SEGMENTS
      GO TO 20
5         CONTINUE
C         CALL DENDAT(BX,BY,RPERX)                  ; DENSITY DATA
          GO TO 30
6         NUM=LIST(NLIST)
          CALL S4EDIT(NUM,BX,BY,RPERX)
          GO TO 30

300         CALL ADIEU(XR,YT)              ; FINISH UP
      GO TO 200

      END
```

```
C        TITLE: GETSEG.FR
C
C    GETSEG DISPLAYS SEGMENTS ON SCREEN.
C
C    CALLING SEQUENCE: CALL GETSEG(IZONE,KSEGS,NSERS)
C           WHERE          IZONE=         ZONE OR ZONES TO BE CONSIDERED
C                           (0=BOTTOM, 3=TOP, 4=0 AND 1, 5=1,2
C                           6=2,3 AND 7=ALL)
C                   NSEGS=        NUMBER OF SEGS TO BE DISPLAYED
C                   NSERS=        ARRAY CONTAINING SEGMENT SERIAL NOS.
C
C    GETSEG MAKES CHECKS AS IT GOES ALONG TRYING TO DETERMINE IF
C    A SEGMENT OR SEGMENT PART IS OUTSIDE THE CURRENT WINDOW.
C    IT CHECKS ON THE SEG'S LAST POINT BEING TO THE LEFT OF
C    SINIST, SEG'S FIRST POINT BEING RIGHT OF DEXTER, SEG'S
C    FIRST Y BEING MUCH TOO FAR AWAY TO EXPECT ANY OF IT TO
C    BE WITHIN TOP AND BOTTOM.
C
C    GETSEG LABELS EACH SEGMENT BEGINNING OR WHERE IT ENTERS
C    THE WINDOW WITH THE SERIAL NO, LABELS A SEGMENT END (BUT
C    NOT WHERE IT LEAVES WINDOW) WITH A SMALL TRIANGLE POINTING
C    TO THE END.  MISSING DATA (SIGNALLED BY Y=0) IS INDICATED
C    BY DOTTED LINES.
C
```

```
C     NOTE THAT IF A PORTION OF A SEGMENT ENDS WITHIN THE WINDOW
C     WITHOUT A TRIANGLE, IT MEANS THAT THERE IS MISSING
C     DATA AND THE NEXT REAL POINT LIES OUTSIDE THE WINDOW.
C
C     GETSEG IS CALLED BY MAIN, MANIPU, DELSEG, FILLIN.
C
C     GETSEG CALLS INSIDE, XEND, MAPRD.
C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C

         SUBROUTINE GETSEG(IZONE,KSEGS,NSERS)

          COMMON/LOCAL/BX(4),BY(4),XPAIR(2),YPAIR(2),XLAST,RPERX
         COMMON/CNSTNT/NPARM,ICH,ISBYTS,IPBYTS,MAXPTS,LCH
      1              ,NWRDS,IRBYTS,IMBYTS,MPWRDS,IFILE(3)
         COMMON/BUFFS/PARAM(20),SEGPTS(64)
         COMMON/SQRD/SINIST,DEXTER,BOTTOM,TOP
         COMMON/RSLTN/IRESX,IRESY
C         DIMENSION COEF(3),XPAIR(2),YPAIR(2),NSERA(3),NSERS(1),NFILE(3)
          DIMENSION COEF(3),NSERA(3),NSERS(1),NFILE(3)
         LOGICAL MOVED,LABELD


          XLASO=XLAST
          NSEGS=IABS(KSEGS)
         NFILE(3)="  "
         NFILE(1)="SG"
         SMALLX=4.*((DEXTER-SINIST)/1023.)
         KEYSAV=0


         DO 1 J=1,NSEGS                ; LOOP ON THE LIST OF SEGS

         NSER1=IAND(NSERS(J),107777K)        ; MASK OFF ZONE
         NSER=NSER1
         IF(IZONE.EQ.7)GO TO 16              ; WANTS ALL ZONES
         ISZONE=IAND(NSERS(J),070000K)/4096
                             ;GET ZONE BITS AND SHIFT RIGHT
         IF(ISZONE.EQ.IZONE)GO TO 16
         IF(IZONE.LT.4)GO TO 1
         IF(ISZONE.EQ.(IZONE-4))GO TO 16
         IF(ISZONE.EQ.(IZONE-3))GO TO 16
         GO TO 1

16         DO 6 K=1,3                ; CONVERT SER. NO. TO ASCII
         L=3-K
         IND=NSER/10**L
         NSERA(K)=IND+48
         NSER=NSER-((10**L)*IND)
6         CONTINUE

         NSER=NSER1
         MOVED=.FALSE.
         LABELD=.FALSE.
         KEYSAV=0
         IPNTR=0
         XPAIR(1)=0.
         YPAIR(1)=0.


22       CALL MAPRD
      1   (NSER,ITYPE,NO,NFILE(2),ISREC,IBEGIN,FIRST,IFREC,LASTP,IPNTR)
C         IF(NO.EQ.0)TYPE "MAP ERROR IN GETSEG"
C         IF(NO.EQ.0)STOP
          IF(NO.GT.0)GO TO 23
          KSEGS=1
```

```
      RETURN
23    CONTINUE

      KEY=2
      IF(FIRST.GE.DEXTER)GO TO 15
      VFIRST=FIRST
      IBEGAN=IBEGIN

      IF(XEND(FIRST,IFREC,ISREC,IBEGIN,LASTP).LE.SINIST)GO TO 25

      KEY=0                   ;END OF THIS PIECE NOT TO LEFT OF WINDOW
      IST=ISREC+1                 ; SEARCH FOR FIRST RECORD
      LST=IFREC+1                 ; FALLING INSIDE WINDOW

      DO 55 KK=IST,LST
      IREC=KK-1
      MAX=MAXPTS
      IF(IREC.EQ.IFREC)MAX=LASTP
      IF(XEND(FIRST,IREC,ISREC,IBEGIN,MAX).LE.SINIST)GO TO 55
      GO TO 56
55      CONTINUE
      STOP 6                  ; CAN NEVER FALL THRU THIS LOOP

56      LPOINT=IBEGIN                  ; SET FIRST WORD USED IN REC
      IF(IREC.NE.ISREC)LPOINT=1
                                      ; SET FIRST X-VALUE
      FIRST=XEND(FIRST,IREC,ISREC,IBEGIN,LPOINT)

                                      ; OPEN THE POINTS FILE AND READ
                                      ; IN A RECORD OF Y-VALUES.
      CALL OPEN(ICH,NFILE,1,IER,ISBYTS)
      X=FIRST
      IFX=IFIX(FIRST)
100      CALL READR(ICH,IREC,SEGPTS,1,IER)

      LPTS=LASTP
      IF(IREC.EQ.ISREC)GO TO 5
      IBEGIN=1
5      IF(IREC.LT.IFREC)LPTS=MAXPTS
                                      ; EVENTUALLY, XPAIR AND YPAIR
                                      ; WILL CONTAIN THE TWO POINTS
                                      ; BRACKETING AN AREA OF MISSING
                                      ; DATA.  A STRAIGHT DOTTED LINE
                                      ; WILL BE DRAWN BETWEEN THEM.
      IF(YPAIR(1).EQ.0.)YPAIR(1)=SEGPTS(IBEGIN)

      DO 1000 L=IBEGIN,LPTS,IRESY         ; LOOP ON POINTS
      Y=SEGPTS(L)
      CALL INSIDE(X,Y,KEY)
      IF(KEY.EQ.3)GO TO 10            ;TOO FAR AWAY IN Y TO CONSIDER
      IF(KEY.EQ.0)GO TO 29                 ; INSIDE WINDOW
      IF(KEY.EQ.1)GO TO 35                 ; MISSING DATA
      IF(KEYSAV.NE.1)GO TO 35
      GO TO 31

29      IF(MOVED)GO TO 30                 ; FIRST POINT?
31      XPAIR(2)=X
      YPAIR(2)=Y
      IF(KEYSAV.EQ.1)GO TO 26                 ; WAS MISSING DATA
      GO TO 28


26      IF(YPAIR(1).EQ.0.)GO TO 28
      IF(YPAIR(2).EQ.0.)GO TO 28
       IF(KSEGS.GE.0.OR.XPAIR(2).LT.XLAST)GO TO 261
       XLAST=XPAIR(2)+1.
```

```
      IF((XLAST-XLASO).LE.1.)KSEGS=1
      IF(XLAST.GE.DEXTER)KSEGS=1
      CALL CLOSE(ICH,IER)
      RETURN
261   CONTINUE

            ; COMPUTE A STRAIGHT LINE BETWEEN
            ; XPAIR(1),YPAIR(1) AND XPAIR(2),YPAIR(2).
            ; MINIMIZE THE VALUES: PFIT IS MORE ACCURATE
            ; WITH SMALL MAGNITUDES.
      XNORM=AMIN1(XPAIR(1),XPAIR(2))
      YNORM=AMIN1(YPAIR(1),YPAIR(2))

      DO 65 IP=1,2
      XPAIR(IP)=XPAIR(IP)-XNORM
65      YPAIR(IP)=YPAIR(IP)-YNORM
      CALL PFIT(XPAIR,YPAIR,2,COEF,IER)
      XDOT=XPAIR(1)

      DO 66 IP=1,2
      XPAIR(IP)=XPAIR(IP)+XNORM
66      YPAIR(IP)=YPAIR(IP)+YNORM

            ; NOW MOVE ACROSS THE MISSING DATA
            ; GAP COMPUTING Y-VALUES FOR THE DOTTED LINE.

27      XDOT=XDOT+SMALLX
      REALX=XDOT+XNORM
      IF(REALX.GE.X)GO TO 28
      YDOT=COEF(1)+(XDOT*COEF(2))+YNORM
      CALL MOVEA(REALX,YDOT)
      CALL INSIDE(REALX,YDOT,KEYSM)
      IF(KEYSM.EQ.0)CALL DRWREL(0,1)
      GO TO 27

28      IF(KEY.EQ.2)GO TO 35
      CALL MOVEA(X,Y)
      MOVED=.TRUE.

30      IF(LABELD)GO TO 45              ; SERIAL NO. WRITTEN?
      CALL ANMODE
      CALL ADEOUT(3,NSERA)
      IF(ITYPE.EQ.0)GO TO 33
      CALL MOVREL(1,0)
      CALL ANMODE
      CALL ADEOUT(3,NSERA)              ; REF LINES IN BOLDFACE
33      CALL MOVEA(X,Y)
      LABELD=.TRUE.
      GO TO 40

45      CALL DRAWA(X,Y)              ; NOT A FIRST POINT: DRAW LINE
      GO TO 40
C
35      MOVED=.FALSE.              ; PRETEND NEXT POINT IS A FIRST
      IF(KEY.EQ.2)LABELD=.FALSE.

40      IFX=IFX+IRESY

      IF(KEY.EQ.1)GO TO 41
      XPAIR(1)=X              ; SAVE POINT IN CASE MISSING DATA
      YPAIR(1)=Y              ; COMING UP.
41      X=FLOAT(IFX)
      KEYSAV=KEY
      IF(X.GT.DEXTER)GO TO 10
1000      CONTINUE
```

```
      X=XEND(VFIRST,IREC,ISREC,IBEGAN,LPTS)+1.                          GSEG0223
      IFX=IFIX(X)                                                       GSEG0224
      IREC=IREC+1                                                       GSEG0225
      IF(IREC.LE.IFREC)GO TO 100        ; MORE POINT BLOCKS?            GSEG0226
10      CALL CLOSE(ICH,IER)                                             GSEG0227
      IF(X.GE.DEXTER)GO TO 1                                            GSEG0228
      IF(KEY.EQ.3)GO TO 1                                               GSEG0229
25      IF(IPNTR.EQ.0)GO TO 15                ; END OF SEG              GSEG0230
      GO TO 22                  ; GO GET NEXT SEGMENT OF TRACE          GSEG0231
                                                                        GSEG0232
15      IF(KEY.NE.0)GO TO 1         ; IF END NOT IN WINDOW, DON'T MARK  GSEG0233
      CALL DRWREL(-3,3)                ; DRAW LITTLE TRIANGLE           GSEG0234
      CALL DRWREL(0,-6)                ; MARKING END OF SEGMENT         GSEG0235
      CALL DRWREL(3,3)                                                  GSEG0236
      LABELD=.FALSE.                                                    GSEG0237
                                                                        GSEG0238
       CONTINUE                                                         GSEG0239
       IF(KSEGS.EQ.-1)KSEGS=1                                           GSEG0240
                                                                        GSEG0241
      RETURN                                                            GSEG0242
      END                                                               GSEG0243
                                                                        GSEG0244
```

```
C        TITLE: BOUND.FR                                                BOND0001
C                                                                       BOND0002
C   CALLING SEQUENCE: CALL BOUND(BNDX,BNDY,JOB)                         BOND0003
C                                                                       BOND0004
C          BNDX, BNDY ARE THE CORNER COORDINATES OF A RECTANGLE MEASURED
                                                                        BOND0005
C          IN THIS ROUTINE BY THE OPERATOR AND DEFINING THE NEXT WINDOW BOND0006
C          OR THE AREA IN WHICH RAW DATA IS TO BE DISPLAYED.            BOND0007
C                                                                       BOND0008
C          JOB=1:  DISPLAY RAW DATA                                     BOND0009
C          JOB=2:  ESTABLISH NEW WINDOW                                 BOND0010
C          JOB=3:  AUTO-DIGITIZATION                                    BOND0011
C          JOB=4:  ACTIVATE MANIPULATION ROUTINES                       BOND0012
C               JOB=5:   DENSITY DATA                                   BOND0013
C               JOB=6:   4-STEPS MENU                                   BOND0014
C                                                                       BOND0015
C   BOUND CALLS SUBROUTINE TELL.                                        BOND0016
C   BOUND IS CALLED BY MAIN.                                            BOND0017
C                                                                       BOND0018
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C
                                                                        BOND0019
                                                                        BOND0020
      SUBROUTINE BOUND(BNDX,BNDY,JOB)                                   BOND0021
                                                                        BOND0022
       COMMON/YCUTS/ICUTS,YCUT(10)                                      BOND0023
C      COMMON/YCUTS/ICUTS,XCUT(4),YCUT(4)                               BOND0024
      COMMON/SQRD/SINIST,DEXTER,BOTTOM,TOP                              BOND0025
      COMMON/BND/MSG(50)                                                BOND0026
      COMMON/MSGS/IVER,IHOR,IX,IY                                       BOND0027
      COMMON/RSLTN/IRESX,IRESY                                          BOND0028
       COMMON/LODAT/MSG0(20),MSG2(15)                                   BOND0029
      DIMENSION BNDX(1),BNDY(1),ICHARS(72),X(4),Y(4)                    BOND0030
       DATA ICUTS,YCUT/0,10*0./                                         BOND0031
      DATA MSG0/34,65,34,85,84,79,45,68,73,71,                          BOND0032
```

```
     1  73,84,73,90,65,84,73,79,78,44/,                                  BOND0033
     1  MSG2/34,52,34,45,83,84,69,80,83,32,77,69,78,85,44/               BOND0034
                                                                         BOND0035
       DATA MSG(1),MSG(2),MSG(3),MSG(4),MSG(5),MSG(6),                   BOND0036
     1 MSG(7),MSG(8),MSG(9),MSG(10),MSG(11),MSG(12),                     BOND0037
     2 MSG(13),MSG(14),MSG(15),MSG(16),MSG(17),MSG(18),                  BOND0038
     3 MSG(19),MSG(20),MSG(21),MSG(22),MSG(23),                          BOND0039
     4 MSG(24),MSG(25),MSG(26),MSG(27),MSG(28),MSG(29),MSG(30),MSG(31),  BOND0040
     5 MSG(32),MSG(33),MSG(34),MSG(35),MSG(36),MSG(37),MSG(38),MSG(39),  BOND0041
     6 MSG(40),MSG(41),MSG(42),MSG(43),MSG(44),MSG(45),                  BOND0042
     7 MSG(46)/                                                          BOND0043
     1             34,82,34,65,87,32,68,65,84,65,  44,32,                BOND0044
     2      34,66,34,76,79,87,45,85,80,  44,32,                          BOND0045
     3      34,80,34,82,79,67,69,68,85,82,69,83,  44,32,                 BOND0046
     4      34,83,34,67,65,76,69, 58,32/                                 BOND0047
                                                                         BOND0048
10      CONTINUE                                                         BOND0049
        CALL TELL(15,MSG2,1)                                             BOND0050
C       CALL TELL(15,MSG1,1)                                             BOND0051
       CALL TELL(20,MSG0,1)              ; DISPLAY OPTIONS               BOND0052
        CALL TELL(23,MSG,1)                                              BOND0053
       CALL TELL(23,MSG(24),2)                                           BOND0054
       JOB=0                                                             BOND0055
                                                                         BOND0056
       LIM=3                                                             BOND0057
       CALL ADEIN(NCHAR,ICHARS)        ; READ REQUEST                    BOND0058
       IF(NCHAR.EQ.0)GO TO 3                                             BOND0059
       KAR=ICHARS(NCHAR)                                                 BOND0060
       IF(KAR.EQ.82)GO TO 1              ; "R"AW DATA                    BOND0061
       IF(KAR.EQ.18)GO TO 1              ; ^R (SAME BOUNDS AS BEFORE)    BOND0062
                                                                         BOND0063
       IF(KAR.EQ.66)GO TO 2              ; "B"LOW-UP                     BOND0064
       IF(KAR.EQ.2) GO TO 2              ; ^B (SAME BOUNDS AS BEFORE)    BOND0065
                                                                         BOND0066
        IF(KAR.EQ.80)GO TO 4              ; "P"ROCEDURES                 BOND0067
                                                                         BOND0068
       IF (KAR.EQ.65)GO TO 5              ; "A"UTO-DIGITIZATION          BOND0069
                                                                         BOND0070
        IF(KAR.EQ.68)GO TO 6                 ; "D"ENSITY DATA            BOND0071
                                                                         BOND0072
        IF(KAR.EQ.52)GO TO 7               ; "4"-STEPS MENU              BOND0073
                                                                         BOND0074
       WIDTH=0.                                                          BOND0075
       HIGH=0.                                                           BOND0076
       IF(KAR.EQ.62)GO TO 46              ; > MOVE TO END OF SEGMENT     BOND0077
                                                                         BOND0078
       IF(KAR.NE.83)GO TO 10              ; "S"CALE                      BOND0079
       LIM=2                                                             BOND0080
       GO TO 11                                                          BOND0081
                                                                         BOND0082
7       JOB=6                                                            BOND0083
        CALL MENU(1,IANS)                                                BOND0084
        IF(IANS.EQ."E")GO TO 10                                          BOND0085
        GO TO 11                                                         BOND0086
6       JOB=5                                                            BOND0087
        GO TO 11                                                         BOND0088
5        JOB=3                                                           BOND0089
       GO TO 11                                                          BOND0090
4       JOB=4                                                            BOND0091
       RETURN                                                            BOND0092
3       JOB=0                                                            BOND0093
       RETURN                                                            BOND0094
2       CONTINUE                                                         BOND0095
       JOB=2                                                             BOND0096
       IF(KAR.EQ.2)RETURN                                                BOND0097
       GO TO 11                                                          BOND0098
```

```
1        CONTINUE
        JOB=1
        IF(KAR.EQ.18)RETURN

11        DO 20 I=1,LIM                ; TAKE 2 OR 3 COORDINATES
        CALL VCURSR(KARC,X(I),Y(I))
        IF(KARC.EQ.63)GO TO 10
        IF(KARC.EQ.32)GO TO 10
        X(I)=FLOAT(IFIX(X(I)))
        IF(I.EQ.1)GO TO 30
        DIF=(X(I)-XLST)**2 + (Y(I)-YLST)**2
        IF(DIF.LT.FLOAT(IRESX))GO TO 10        ; DON'T ALLOW ZERO LENGTH
        IF(KAR.EQ.83)GO TO 25              ; "S"CALE REQUEST
        CALL DRAWA(X(I),Y(I))
        GO TO 20
30        XLST=X(I)
        YLST=Y(I)
        CALL MOVEA(XLST,YLST)
20        CONTINUE

        GO TO 21
25        XN=X(1)
        YN=Y(1)
        XL=X(2)
        KNT=10

        DO 22 J=1,10000              ; DRAW A SCALE
        IUP=3
        IF(KNT.NE.10)GO TO 23
        IUP=6
        KNT=0
23        CALL MOVEA(XN,YN)
        CALL DRWREL(0,IUP)        ; TIC MARK
        KNT=KNT+1
        CALL ANMODE
        IF(J.EQ.1)WRITE(10,999) XN,YN        ; DISPLAY COORDINATE
999        FORMAT(1H ,F5.0,",",F7.2)
        XN=XN+1.
        IF(XN.GT.XL)GO TO 10
        IF(J.NE.1)GO TO 22
        CALL ANMODE
22        CONTINUE

21        CALL MOVEA(X(1),Y(1))          ; DRAW WINDOW OUTLINE
        BNDX(4)=X(1)+(X(3)-X(2))
        BNDY(4)=Y(1)+(Y(3)-Y(2))
        CALL DRAWA(BNDX(4),BNDY(4))
        CALL DRAWA(X(3),Y(3))
        DO 35 J=1,3
        BNDX(J)=X(J)
        BNDY(J)=Y(J)
35        CONTINUE
         ICUTS=0
         IF(KARC.NE.44)RETURN
         IF(JOB.NE.3.AND.JOB.NE.6)RETURN
         ICUT=0
201      CONTINUE
         CALL VCURSR(KARC,XMN1,YMN1)
         CALL MOVEA(BNDX(1),YMN1)
         CALL DRAWA(BNDX(3),YMN1)
         CALL VCURSR(KARC,XMN2,YMN2)
         CALL MOVEA(BNDX(1),YMN2)
         CALL DRAWA(BNDX(3),YMN2)
         ICUT=ICUT+1
         YCUT(ICUT)=AMIN1(YMN1,YMN2)
         ICUT=ICUT+1
```

```
        YCUT(ICUT)=AMAX1(YMN1,YMN2)
        IF(KARC.EQ.44)GO TO 201
        ICUTS=ICUT
C       DO 201 I=1,3
C       CALL VCURSR(KARC,X(I),Y(I))
C       IF(I.GT.1)GO TO 202
C       CALL MOVEA(X(1),Y(1))
C       GO TO 201
C202    CALL DRAW(X(I),Y(I))
C201    CONTINUE
C       X(4)=X(1)+X(3)-X(2)
C       Y(4)=Y(1)+Y(3)-Y(2)
C       CALL DRAWA(X(4),Y(4))
C       CALL DRAWA(X(1),Y(1))
       RETURN

               ; FOLLOWING MOVES CURRENT WINDOW UP, DOWN
               ; LEFT, OR RIGHT, WITH A 10% OVERLAP

               ; MOVE TO END OF SEGMENT

46        WIDTH=DEXTER-SINIST
       HIGH=TOP-BOTTOM

       CALL ENDSPOT(ENDX,ENDY,IER)
       IF(IER.NE.1)GO TO 10

               ; SET WINDOW AT CURRENT DIMENSIONS
               ; CENTERED ON X,Y OF END OF SEGMENT

       BNDX(1)=ENDX-(WIDTH/2.)
       BNDX(2)=ENDX+(WIDTH/2.)
       BNDX(3)=BNDX(2)
       BNDX(4)=BNDX(1)
       BNDY(1)=ENDY-(HIGH/2.)
       BNDY(2)=BNDY(1)
       BNDY(3)=ENDY+(HIGH/2.)
       BNDY(4)=BNDY(3)
       JOB=2
       RETURN
       END

*)
```

```
      SUBROUTINE S4EDIT(NUM,BX,BY,RPERX)                                EDIT0001
      DIMENSION BX(1),BY(1)                                             EDIT0002
      COMMON/RSLTN/IRESX,IRESY                                          EDIT0003
      COMMON/LOS/NSEGS,NSER(500),NLIST,LIST(5)                          EDIT0004
      NUM1=NUM                                                          EDIT0005
      XL=AMIN1(BX(1),BX(2),BX(3),BX(4))                                 EDIT0006
      XR=AMAX1(BX(1),BX(2),BX(3),BX(4))+IRESX                           EDIT0007
      MUM1=NUM1                                                         EDIT0008
      CALL SPLIT(XL,MUM1)                                               EDIT0009
      CALL GETSEG(7,1,MUM1)                                             EDIT0010
      NUM2=MUM1                                                         EDIT0011
      MUM2=NUM2                                                         EDIT0012
      CALL SPLIT(XR,MUM2)                                               EDIT0013
      CALL GETSEG(7,1,MUM2)                                             EDIT0014
      NUM3=MUM2                                                         EDIT0015
      CALL GETSEG(7,1,NUM2)                                             EDIT0016
      CALL DELSEG(NUM2)                                                 EDIT0017
      CALL AUTODG(BX,BY,RPERX,NUM2)                                     EDIT0018
      NL=NLIST                                                          EDIT0019
      L1=LIST(1)                                                        EDIT0020
      L2=LIST(2)                                                        EDIT0021
      L3=LIST(3)                                                        EDIT0022
      L4=LIST(4)                                                        EDIT0023
      L5=LIST(5)                                                        EDIT0024
      NLIST=3                                                           EDIT0025
      LIST(1)=NUM1                                                      EDIT0026
      LIST(2)=NUM2                                                      EDIT0027
      LIST(3)=NUM3                                                      EDIT0028
      CALL MERGE(NUM4)                                                  EDIT0029
      CALL GETSEG(7,1,NUM4)                                             EDIT0030
      NLIST=NL                                                          EDIT0031
      LIST(1)=L1                                                        EDIT0032
      LIST(2)=L2                                                        EDIT0033
      LIST(3)=L3                                                        EDIT0034
      LIST(4)=L4                                                        EDIT0035
      LIST(5)=L5                                                        EDIT0036
      RETURN                                                            EDIT0037
      END                                                               EDIT0038
                                                                        EDIT0039
```

```
C                                                                       MENU0001
C                 TITLE: MENU.FR                                        MENU0002
C                                                                       MENU0003
C       MENU REQUEST INFORMATION TO DISPLAY SOME OR ALL OF THE          MENU0004
C       SEGMENTS, OR TO QUIT.                                           MENU0005
C                                                                       MENU0006
C       MENU CALLS TELL, ADEIN, ADEOUT.                                 MENU0007
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C MENU0008
C                                                                       MENU0009
      SUBROUTINE MENU(INIT,IANS,IZONE)                                  MENU0010
                                                                        MENU0011
      COMMON/LOS/NSEGS,NSER(500),NLIST,LIST(5),NSWER                    MENU0012
      COMMON/MSGS/IVER,IHOR,IX,IY                                       MENU0013
      DIMENSION KARS(72)                                                MENU0014
      COMMON/DATAS/MSG(50),MSGA(40),MSG2(21)                            MENU0015
        DATA MSG2                                                       MENU0016
     1 /67,72,69,67,75,32,70,73,76,76,73,78,63,32,40,89,47,78,41,58,32/ MENU0017
```

```
      DATA MSG(1),MSG(2),MSG(3),MSG(4),MSG(5),MSG(6),MSG(7),MSG(8),
     1  MSG(9),MSG(10),MSG(11),MSG(12),MSG(13),MSG(14),MSG(15),
     2  MSG(16),MSG(17),MSG(18),MSG(19),MSG(20),MSG(21),MSG(22),
     4  MSG(23),MSG(24),MSG(25),MSG(26),MSG(27),
     5  MSG(28),MSG(29),MSG(30),MSG(31),MSG(32),
     6  MSG(33),MSG(34),MSG(35),MSG(36),MSG(37),MSG(38),MSG(39),
     7  MSG(40),MSG(41),MSG(42),MSG(43),MSG(44),MSG(45),MSG(46),
     8  MSG(47),MSG(48),MSG(49),MSG(50)/

     1       68,73,83,80,76,65,89,32,34,65,34,76,76,32,
     1       83,69,71,77,69,78,84,83,44,
     2       34,83,34,79,77,69,44,32,79,82,32,34,78,34,79,78,69,58,
     3       32,32,83,69,71,58,
     4       63,58,48/

      DATA MSGA(1),MSGA(2),MSGA(3),MSGA(4),MSGA(5),MSGA(6),MSGA(7),
     1  MSGA(8),MSGA(9),MSGA(10),MSGA(11),MSGA(12),MSGA(13),
     2  MSGA(14),MSGA(15),MSGA(16),MSGA(17),MSGA(18),MSGA(19),
     3  MSGA(20),MSGA(21),MSGA(22),MSGA(23),MSGA(24),MSGA(25),MSGA(26),
     4  MSGA(27),MSGA(28),MSGA(29),MSGA(30),MSGA(31),MSGA(32),MSGA(33),
     5  MSGA(34),MSGA(35),MSGA(36),MSGA(37),MSGA(38),MSGA(39),MSGA(40)/

     1  90,79,78,69,58,32,48,32,49,32,50,32,51,         ; ZONE: 0 1 2 3
     2  32,32,32,32,32,32,32,52,32,53,32,54,            ;         4 5 6
     3  32,32,32,32,32,32,32,32,32,55,32,32,32,58,32/    ;       7   :


      NLIST=0
1      IF(INIT.EQ.1)GO TO 100


      CALL TELL(23,MSG,1)             ; DISPLAY "A"LL SEGMENTS,
      CALL TELL(18,MSG(24),2)            ; "S"OME, OR "N"ONE:
      CALL ADEIN(NCHAR,KARS)            ; READ ANSWER A, S, OR N
      IF(NCHAR.EQ.0)GO TO 300            ; QUIT
        IF(KARS(NCHAR).EQ.78)GO TO 400         ;DISPLAY NONE
        IF(KARS(NCHAR).EQ.65)GO TO 10       ; DISPLAY ALL SEGMENTS
      IF(KARS(NCHAR).NE.83)GO TO 1  ;DISPLAY SOME (ANYTHING ELSE ILLEGAL)


        CALL TELL(21,MSG2,1)
        CALL ADEIN(NCHAR,KARS)
        NSWER=KARS(NCHAR)

      IZONE=7

100       DO 30 J=1,5                      ; MAXIMUM FIVE SEGMENTS
      CALL TELL(6,MSG(42),1)                 ; SEG:
31      CALL ADEIN(NCHAR,KARS)                 ; READ THREE DIGITS
      IF(NCHAR.EQ.3)GO TO 32
      IF(NCHAR.EQ.0)GO TO 33                ; DONE
34      CALL ADEOUT(2,MSG(48))                 ; ?:
      GO TO 31
32      IF(KARS(NCHAR).EQ.63)GO TO 200        ; WANTS TO GET OUT
      NUM=3
      CALL CTON(KARS,NUM)              ; CONVERT CHARACTER TO NUMBER
      IF(NUM.EQ.1000)GO TO 34
      LIST(J)=NUM
      NLIST=J
30      CONTINUE

33      IF(NLIST.EQ.0)GO TO 1
      DO 40 J=1,NLIST
      NSG=LIST(J)
      DO 41 K=1,NLIST
      IF(K.EQ.J)GO TO 41
```

```
      IF(LIST(K).EQ.NSG)GO TO 200        ; ERROR: DUPLICATE
41      CONTINUE
      DO 42 K=1,NSEGS
      IF(NSG.EQ.IAND(NSER(K),107777K))GO TO 40
42      CONTINUE
      GO TO 200                     ; ERROR: NOT IN LIST
40      CONTINUE

      IANS="S"
      RETURN
10      IANS="A"
      CALL TELL(13,MSGA,1)              ; ZONE: 0 1 2 3
      CALL TELL(12,MSGA(14),1)       ;         4 5 6
      CALL TELL(15,MSGA(26),2)       ;          7   :
      CALL ADEIN(NCHAR,KARS)
      IF(NCHAR.EQ.0)GO TO 1
      IZONE=KARS(NCHAR)-48
      IF((IZONE.LT.0).OR.(IZONE.GT.7))GO TO 10
      RETURN
200     IANS="E"
      RETURN
300     IANS="Q"
      RETURN
400     IANS="N"
      RETURN

      END
```

```
      SUBROUTINE CHKSEG(IZONE,KSEGS,NSERS)
      COMMON/RSLTN/IRESX,IRESY
      COMMON/LOS/NSEGS,NSER(500),NLIST,LIST(5)
      COMMON/SQRD/SINIST,DEXTER,BOTTOM,UP
      COMMON/LOCAL/CX(4),CY(4),XPAIR(2),YPAIR(2),XLAST,RPERX
      DIMENSION BX(4),BY(4),BXS(4),BYS(4),NSERS(1)
      DO 1000 NN=1,KSEGS
      XLAST=SINIST
      NS=NSERS(NN)
      DO 2 J=1,4
      BX(J)=CX(J)
      BY(J)=CY(J)
      BXS(J)=BX(J)
      BYS(J)=BY(J)
2     CONTINUE
1     CONTINUE
      KSEG=-1
      IF((DEXTER-XLAST).LE.5.)KSEG=1
      CALL GETSEG(IZONE,KSEG,NS)
      IF(KSEG.EQ.1)GO TO 991
      BX(1)=AMAX1(SINIST,XPAIR(1)-5.)
      BY(1)=AMAX1(YPAIR(1),YPAIR(2))+20.
      BY(1)=AMIN1(BY(1),UP)
      BX(2)=BX(1)
      BY(2)=AMIN1(YPAIR(1),YPAIR(2))-20.
      BY(2)=AMAX1(BY(2),BOTTOM)
      BX(3)=AMIN1(DEXTER,XPAIR(2)+5.)
      BY(3)=BY(2)
```

```
      BX(4)=BX(3)                                                       KSEG0029
      BY(4)=BY(1)                                                       KSEG0030
      IF(ABS(BX(3)-BX(1)).LE.IRESX)GO TO 991                            KSEG0031
      IF(ABS(BY(1)-BY(2)).LE.IRESY)GO TO 991                            KSEG0032
      CALL MAGFY(BX,BY)                                                 KSEG0033
      CALL GETSEG(IZONE,NSEGS,NSER)                                     KSEG0034
      CALL RAWDAT(BX,BY,RPERX)                                          KSEG0035
      CONTINUE                                                          KSEG0036
20    CONTINUE                                                          KSEG0037
      DO 205 J=1,4                                                      KSEG0038
      BXS(J)=BX(J)                                                      KSEG0039
205   BYS(J)=BY(J)                                                      KSEG0040
207   CALL BOUND(BX,BY,JOB)                                             KSEG0041
      IF(JOB.EQ.0)GO TO 80                                              KSEG0042
      GO TO (21,31,41,51,61,71),JOB                                     KSEG0043
21    CALL RAWDAT(BX,BY,RPERX)                                          KSEG0044
      GO TO 73                                                          KSEG0045
31    CALL MAGFY(BX,BY)                                                 KSEG0046
      CALL GETSEG(IZONE,NSEGS,NSER)                                     KSEG0047
      GO TO 20                                                          KSEG0048
41    CALL AUTODG(BX,BY,RPERX,NUM)                                      KSEG0049
      GO TO 73                                                          KSEG0050
51    CALL MANIPU                                                       KSEG0051
      GO TO 20                                                          KSEG0052
61    CONTINUE                                                          KSEG0053
C     CALL DENDAT(BX,BY,RPERX)                                          KSEG0054
      GO TO 73                                                          KSEG0055
71    NUM=LIST(NLIST)                                                   KSEG0056
      CALL S4EDIT(NUM,BX,BY,RPERX)                                      KSEG0057
73    DO 75 J=1,4                                                       KSEG0058
      BX(J)=BXS(J)                                                      KSEG0059
75    BY(J)=BYS(J)                                                      KSEG0060
      GO TO 20                                                          KSEG0061
80    CONTINUE                                                          KSEG0062
      DO 81 J=1,4                                                       KSEG0063
      BX(J)=CX(J)                                                       KSEG0064
81    BY(J)=CY(J)                                                       KSEG0065
      CALL MAGFY(BX,BY)                                                 KSEG0066
      GO TO 1                                                           KSEG0067
991   CONTINUE                                                          KSEG0068
1000  CONTINUE                                                          KSEG0069
      RETURN                                                            KSEG0070
      END                                                               KSEG0071
                                                                        KSEG0072
```

A.3 KSCRIBE

The final stage of ARDS, after TV, is SCRIBE. The program "SCRIBE" (Trifunac and Lee, 1979) performs the reconstruction, plots the data and/or writes onto a magnetic tape. "SCRIBE" moves through a segment in the first section correcting each coordinate for whatever fiducial skew may be present. When the end of the segment in the section is reached, "SCRIBE" identifies the matching segment in the next section performing the necessary corrections (translations and rotations) to each coordinate of that segment in the next section. This continues until the last section is reached, thus recreating the continuous curve out of two or more digitized sections. This is done for each curve and reference lines present on the original record.

An updated version of the program "SCRIBE" has been developed to allow the user to plot and/or write the raw digitized data onto a disk file instead of writing the data on a file on magnetic tape. This has the advantage of allowing the data to be printed on paper and/or later be transferred to a file on a magnetic tape with arbitrary format, record length or block size.

"KSCRIBE" runs essentially the same way as "SCRIBE". The user begins by typing the name of the program

```
) KSCRIBE ↵
```

and "KSCRIBE" will respond by requesting the mapfile suffix:

```
ENTER MAPFILE SUFFIX: MP____
```

The user responds by typing in the suffix, e.g. 01. "KSCRIBE" then requests the options of plotting and/or writing onto disk:

"P"LOT, "D"ISK, or "B"OTH:

The plotter will be initialized if plotting is requested. If writing onto disk is requested, "KSCRIBE" will open the file "VOX01.DT" on disk as the raw data file of the digitized record. Subsequently, "KSCRIBE" will continue execution in the manner identical to that of "SCRIBE" (Trifunac and Lee, 1979).

The output of the program now consists of the plot of the digitized data in a scale identical to that of the original record, and/or a disk file of the data: "VOX01.DT". The data will be in integer format in units of microns (10^{-4} cm).

The following pages represent a listing of the main program and updated subroutines:

KSCRIBE.FR
TODISK.FR

```
C       TITLE: KSCRIBE.FR
C
C FOR ONE OR MORE SECTIONS & 1 PAIR OF FIDUCIALS/SECTION
C
C ONCE THE TRACES HAVE BEEN COMPLETELY EDITED, THEY CAN BE WRITTEN
C TO DSIK AND, IF NEED BE, PLOTTED WITH SCRIBE.  ALL COORDINATES
C ARE CONVERTED TO MICRONS, ROTATED SUCH THAT THE Y-COORDINATES
C OF THE LEFT AND RIGHT FIDUCIAL PAIRS ARE THE SAME, AND FLAGGED IF
C THEY WERE GENERATED BY THE OPERATOR USING THE GRAPHICAL EDITOR OR
C WERE CALCULATED AS THE STRAIGHT LINE BRIDGING A SECTION OF
C "MISSING" DATA..
C (Y-COORDINATE SET TO ZERO EITHER BY TRACE OR BY TV).
C
C THE TRACES ARE CONCATENATED IF THE ORIGINAL FILM WAS
C "STACKED" ON THE 10X10-INCH NEGATIVE.  THE Y-VALUES ARE
C NORMALIZED TO CAUSE THE FIDUCIALS TO BE CENTERED HALFWAY UP THE
C 11-INCH PLOTTING PAPER.  THIS NORMALIZATION
C IS RETAINED IN THE COORDINATES AS WRITTEN TO DSIK.
C
C WHEN ALL FOUR ZONES OF THE 10X10 NEGATIVE ARE USED, ONE OF THE
C REFERENCE LINES MAY NOT FIT WITHIN THE DENSITOMETER'S RANGE
C IN Y AND NOT APPEAR IN THE BOTTOM ZONE.  IF THIS IS THE CASE,
C THE PROGRAM WILL COMPLAIN THAT IT CAN'T MATCH A SEGMENT TO
C THE CURVE IN ZONE 4.  THE OPERATOR CAN REQUEST THAT DUMMY DATA
C BE INSERTED FOR THE CURVE IN THAT ZONE.
C
C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C

      PARAMETER AMMPI=25.4        ; MM PER INCH
      PARAMETER AMMPM=.001        ; MM PER MICRON

C       PROGRAM SCRIBE

      COMMON/PLOTAP/TAPOUT,PLOUT,PXFACT,PYFACT,XFACT,YFACT,ABBREV
      COMMON/COREKT/ITYPE(40),NSERS(40),IZONE(40),
     1         XFIDL(50),YFIDL(50),XFIDR(50),YFIDR(50),IFIDZ(50),
     2                    XMAX,YMAX,NFIDS,NZONE
      COMMON/TMAP/MAP(504),MAPW,MREC,LSTREC,ICH
      COMMON/SQRD/SINIST,DEXTER,BOTTOM,TOP,HALFUP
      COMMON/PSQRD/PSIN,PDEX,PBOT,PTOP
      COMMON/MAGTAP/NCURV,INITL,LASTC,KOUNT,MISSING,FACTOR(2),
     1           NTYPE
      COMMON/KONKAT/MATCH,MATCHED,CONBUF(2,64),MBUFPT,DEVSQ,CNTDEV
      COMMON/DATAT/KTYPES(4)
      COMMON/EYEOH/LCH,KCH

      DIMENSION INFILE(4),MFILE(3),IDUM(22),MTPFL(3)
      LOGICAL TAPOUT,PLOUT,MATCH,INITL,HEAD,REFS,ABBREV,NEXT

      DATA KTYPES(1),KTYPES(2),KTYPES(3),KTYPES(4)/3,2,1,0/

      LCH=1                ; TPFD INPUT, DISK FILE OUTPUT
      ICH=2                ; MAPFILE INPUT
      KCH=3                ; NOT USED
      SINIST=0.                ; PLOTTER "WINDOW" DIMENSIONS
      DEXTER=50.                ; 50 INCHES IN X, 11 IN Y
      BOTTOM=0.
      TOP   =11.0

      INITL=.TRUE.
      MATCH=.FALSE.
      HEAD=.FALSE.
      LASTC=0
      KOUNT=1
```

```
      INFILE(1)="TP"
      INFILE(2)="FD"
      INFILE(4)="  "
      MFILE(1)="MP"
      MFILE(3)="  "
      MTPFL(1)="MT"
      MTPFL(2)="0:"
      MTPFL(3)="  "

      WRITE(10,300)
300      FORMAT(" ENTER MAPFILE SUFFIX: MP",Z)

      READ(11,301) MFILE(2)
301      FORMAT(A2)

      INFILE(3)=MFILE(2)

      CALL STAT(MFILE,IDUM,IER)
      IF(IER.NE.1)TYPE "MAPFILE NOT FOUND"
      IF(IER.NE.1)STOP
      CALL OPEN(ICH,MFILE,1,IER,1008)
      IF(IER.NE.1)TYPE "MAPFILE ERROR=",IER

      CALL SMAPMAK             ; READ THE MAPFILE

      CALL OPEN(LCH,INFILE,1,IER)
      IF(IER.NE.1)TYPE "TPFD ERROR=",IER

                         ; READ TPFD FILE
      READ(LCH,2)NZONE,NFIDS,NSEGS
      IF(NFIDS.LT.1)GO TO 4
      DO 1 J=1,NFIDS
      READ(LCH,3)XFIDL(J),YFIDL(J),XFIDR(J),YFIDR(J),IFIDZ(J)
1      CONTINUE
2      FORMAT(3I3)
3      FORMAT(4F10.1,I3)

4      DO 7 J=1,NSEGS
      READ(LCH,6) ITYPE(J),NSERS(J),IZONE(J)
6      FORMAT(I1,I4,I2)
      IZONE(J)=NZONE-IZONE(J)        ; CHANGE 0,1,2,3 TO 4,3,2,1
7      CONTINUE

      READ(LCH,3) XMAX,YMAX

      CALL CLOSE(LCH,IER)


      IREC=0
      INITL=.TRUE.
      TAPOUT=.TRUE.
      PLOUT =.TRUE.

      WRITE(10,8)
8      FORMAT(" 'P'LOT, 'D'ISK, OR 'B'OTH:",Z)

      READ(11,9) IANS
9      FORMAT(A2)

      IF(IANS.EQ.2HP )TAPOUT=.FALSE.
      IF(IANS.EQ.2HD )PLOUT=.FALSE.

      IF(.NOT.TAPOUT)GO TO 200

C10      TYPE "MOUNT WRITE-ENABLED TAPE ON MT0"
C      PAUSE
```

```
C                                                                        KSCR0133
C      WRITE(10,11)                                                      KSCR0134
C11      FORMAT(1X" INPUT FILE NO. TO BE WRITTEN (A2) MT0:"Z)            KSCR0135
C      READ(11,9)MTPFL(3)                                                KSCR0136
C      CALL INIT("MT0",0,IER)                                            KSCR0137
C      IF((IER.NE.1).AND.(IER.NE.40))GO TO 100                           KSCR0138
C      IF(IER.EQ.40)TYPE "MT0 ALREADY INIT"                              KSCR0139
C      IF(IER.EQ.40)CALL RLSE("MT0",IERR)                                KSCR0140
C      IF(IER.NE.1)GO TO 100                                             KSCR0141
C                                                                        KSCR0142
C      CALL MTOPD(LCH,MTPFL,0,IER)                                       KSCR0143
C                             ; OPEN MAGTAPE CHANNEL FOR WRITING         KSCR0144
C                                 ; NON-STANDARD RECORD LENGTHS          KSCR0145
C      IF(IER.NE.1)TYPE "MT0 ERROR=",IER                                 KSCR0146
C      GO TO 200                                                         KSCR0147
C                                                                        KSCR0148
C100       TYPE "CAN'T INITIALIZE MT0!"                                  KSCR0149
C      TYPE "INIT IER=",IER                                              KSCR0150
C      GO TO 10                                                          KSCR0151
        CALL DFILW("RAW.DT",IER)        ; DELETE OLD FILE                KSCR0152
        IF((IER.NE.1).AND.(IER.NE.13))STOP FILE PROTECTED                KSCR0153
        CALL OPEN(LCH,"DPOF:V0X01.DT",3,IER)                             KSCR0154
                                                                         KSCR0155
200       NCURV=1                                                        KSCR0156
                                                                         KSCR0157
      IF(PLOUT)CALL INITIAL(6,200,TOP,DEXTER)                            KSCR0158
      WRITE(10,201)                                                      KSCR0159
201       FORMAT(" ENTER DENSITOMETER X- AND Y-RASTER",/,                KSCR0160
     1            " SIZES IN MICRONS: ",Z)                               KSCR0161
                            ; READ IN MICRONS PER UNIT                   KSCR0162
      READ(11) XFACT,YFACT                                               KSCR0163
                            ; ALLOW SKIPPING REFS ON PLOTS               KSCR0164
      WRITE(10,202)                                                      KSCR0165
202       FORMAT(" INCLUDE REFERENCE LINES? (Y/N):",Z)                   KSCR0166
      READ(11,9) IANS                                                    KSCR0167
      REFS=.FALSE.                                                       KSCR0168
      IF(IANS.EQ."Y ")REFS=.TRUE.                                        KSCR0169
C       WRITE(10,203)                                                    KSCR0170
C203     FORMAT(" JUST ENDPOINTS-ONLY VERSION OF DASHED LINES? (Y/N):",Z)
                                                                         KSCR0171
C       READ(11,9) IANS                                                  KSCR0172
C       ABBREV=.FALSE.                                                   KSCR0173
C       IF(IANS.EQ."Y ")ABBREV=.TRUE.                                    KSCR0174
                                                                         KSCR0175
      PXFACT=XFACT*AMMPM*(1./AMMPI)                                      KSCR0176
                ; CONVERTS DENS. UNITS TO INCHES                         KSCR0177
      PYFACT=YFACT*AMMPM*(1./AMMPI)        ; FOR PLOTTING                KSCR0178
      FACTOR(1)=XFACT                                                    KSCR0179
      FACTOR(2)=YFACT                                                    KSCR0180
      PSIN=SINIST*(1./PXFACT)        ; PLOTTER WINDOW IN DENS. UNITS     KSCR0181
      PDEX=DEXTER*(1./PXFACT)                                            KSCR0182
      PBOT=BOTTOM*(1./PYFACT)                                            KSCR0183
      PTOP=TOP    *(1./PYFACT)                                           KSCR0184
                                                                         KSCR0185
      CALL CORECT                         ; SET UP ROTATION AND          KSCR0186
                                  ; CONCATENATION CONSTANTS              KSCR0187
                                                                         KSCR0188
      IF(.NOT.PLOUT)GO TO 222                                            KSCR0189
      CALL PLOT(0.,0.,3)                                                 KSCR0190
      CALL PLOT(1.,0.,2)                                                 KSCR0191
      CALL PLOT(0.,0.,2)                                                 KSCR0192
      CALL PLOT(0.,1.,2)                                                 KSCR0193
                                                                         KSCR0194
      CALL PLOT(0.,0.,2)                                                 KSCR0195
222       MATCHED=0                                                      KSCR0196
      KZONE=1                                                            KSCR0197
```

```
      DO 400 KT=1,4                         ; LOOP ON CURVE TYPES

      IF((.NOT.REFS).AND.(KTYPES(KT).GT.0))GO TO 400

510      DO 401 J=1,NSEGS                   ; LOOP ON SEGMENTS
      IF(IZONE(J).NE.1)GO TO 401                   ; NOT START OF CURVE?
      IF(ITYPE(J).NE.KTYPES(KT))GO TO 401              ; WRONG TYPE?
      IF(NSERS(J).LT.1)GO TO 401                   ; ALREADY DONE?
      NS=J
      TYPE "*************"
      TYPE " CURVE",NCURV
      TYPE "*************"
        WRITE(10,903)NSERS(NS),ITYPE(NS)
903     FORMAT(9X"SEGMENT",I4,", TYPE",I2/
     1  3X,"THIS SEGMENT TO BE SKIPPED (Y/N)? "Z)
        READ(11,9)IANS
        NEXT=.FALSE.
        IF(IANS.EQ."Y ")NEXT=.TRUE.
        IF(NEXT)GO TO 401
      GO TO 501
401      CONTINUE

      GO TO 400

501      MATCHED=0
      MATCH=.FALSE.
      MBUFPT=0
      NTYPE=ITYPE(NS)
      WRITE(10,904) NSERS(NS),NTYPE,KZONE
904   FORMAT(20X,"DOING SEGMENT NUMBER",I4,", TYPE",I2,", IN ZONE",I2)

      CALL SGETSEG(NSERS(NS),KZONE)     ; PLOT AND/OR WRITE THE SEGMENT

      LTYPE=ITYPE(NS)
      NSERS(NS)=-NSERS(NS)              ; FLAG TO NOT USE AGAIN

490      KZONE=KZONE+1
      IF(KZONE.GT.NZONE)GO TO 502
      MATCH=.TRUE.

                        ; LOOK FOR SEGMENT IN NEXT ZONE TO MATCH
      DO 500 J=1,NSEGS
      IF(ITYPE(J).NE.LTYPE)GO TO 500        ; MUST BE SAME TYPE
      IF(IZONE(J).NE.KZONE)GO TO 500        ; MUST BE IN NEXT ZONE
      IF(NSERS(J).LT.1)GO TO 500       ; CAN'T HAVE ALREADY BEEN USED
      CNTDEV=0.
      DEVSQ=0.
      MATCHED=0
      TYPE "     SEGMENT",NSERS(J),":"
      CALL SGETSEG(NSERS(J),KZONE)       ; EXAMINE BEGINNING OF SEG
      IF(MATCHED.NE.1)GO TO 500      ; MATCH UP?
      NS=J                          ; YES!
      GO TO 501
500      CONTINUE

                        ; NOTHING MATCHES
505      NMIS=-NSERS(NS)
      WRITE(10,900) NMIS,KZONE
900      FORMAT(" CAN'T MATCH SEGMENT ",I3," IN ZONE",I2,":")
      IF(.NOT.TAPOUT)GO TO 490
      WRITE(10,901)
901      FORMAT(" 'S'TOP, 'Q'UIT FOR NEXT CURVE (TYPE 2 OR 3):",Z)
      READ(11,902)IANS
902      FORMAT(A2)
      IF(IANS.EQ."S ")STOP              ; NEEDS SOME EDITING
```

```
      IF(ITYPE(NS).LT.2)GO TO 505

      CALL FAKE(KZONE,XFACT)                    ; PUT IN DUMMY DATA

      INITL=.TRUE.
      LASTC=0
      KOUNT=1
      GO TO 490
502      NCURV=NCURV+1
      KZONE=1
      CALL PLOT(0.,0.,3)
      GO TO 510

400      CONTINUE

      IF(.NOT.TAPOUT)STOP
      NCURV=0
      CALL TODISK(0.,0.,3)          ; WRITE OUT THE BUFFER
C      CALL MTDIO(LCH,060000K,IDUM,ISTAT,IER)        ; END OF FILE
C      CALL MTDIO(LCH,010000K,IDUM,ISTAT,IER)        ; REWIND
C      CALL RLSE("MTO",IERR)
        CALL CLOSE(LCH,IER)
      CALL CLOSE(ICH,IER)

      END
```

```
      SUBROUTINE TODISK(X,Y,KPEN)

        COMMON/SHIFT/IXSHFT,IYSHFT
      COMMON/PLOTAP/TAPOUT,PLOUT,PXFACT,PYFACT,XFACT,YFACT,ABBREV
      COMMON/MAGTAP/NCURV,INITL,LASTC,KOUNT,MISSING,FACTOR(2),
     1         ITYPE
      COMMON/LOCAL/XCORZL,YCORZL,XCORZR,YCORZR
      COMMON/EYEOH/LCH,KCH
      COMMON/BUF/ IBUF(1024),POINT(2),LTYPE
      LOGICAL TAPOUT,PLOUT,HEAD,INITL,MISSING,ABBREV
      DATA POINT(1),POINT(2)/0.,0./

      IF(KPEN.EQ.10)GO TO 30                ; FAKE IS IN CONTROL
      IPEN=KPEN
      IF(KPEN.GT.3)IPEN=KPEN-2
      IF(NCURV.EQ.LASTC)GO TO 50          ; STILL ON SAME CURVE

      HEAD=.TRUE.                            ; NEW CURVE
      IF(INITL)GO TO 80

30      KSAV=KOUNT
      DO 40 J=1,2                            ; FLAG END OF CURVE
      IBUF(KOUNT)=177777K
      KOUNT=KOUNT+1
40      CONTINUE
      POINT(1)=-1.
      GO TO 70

50      IF((X-POINT(1))*FACTOR(1).LT.1.)RETURN
              ; COORDINATE MIGHT BE OVERLAPPING
                ; WITH PREVIOUS SEGMENT: DON'T STORE UNTIL
```

```
                    ; WE'VE MOVED BEYOND AREA OF OVERLAP
      POINT(1)=X
      POINT(2)=Y
      MBIT=0
      IF(MISSING)MBIT=ISBIT       ; FLAG "MISSING" DATA
      KSAV=KOUNT

                   ; STORE AS 32-BIT INTEGERS
59      CONTINUE
C     DO 60 J=1,2                        ; LOOP ON X, Y
C
C     PT=POINT(J)*FACTOR(J)       ; CHANGE RASTERS TO MICRONS
C     WLEFT=PT/TTTS                ; LEFT SIDE OF 32-BIT WORD
C     ILEFT=IFIX(WLEFT)
C     WRIGHT=PT-(TTTS*FLOAT(ILEFT))   ;RIGHT SIDE IS MOD(POINT,131072)
C
C     IBUF(KOUNT)=IOR(ILEFT,MBIT)       ; OR IN "MISSING" DATA FLAG
C     ISB=0
C     IF(WRIGHT.LT.TTTF)GO TO 61
C     ISB=ISBIT              ; RIGHT SIDE GT 2**15 -1
C     WRIGHT=WRIGHT-TTTF        ; MAKE IT LESS FOR IFIX
C61     IWRITE=IFIX(WRIGHT)
C     IBUF(KOUNT+1)=IOR(IWRITE,ISB)       ; OR 2**14 BACK IN
C
C     KOUNT=KOUNT+2
C60     CONTINUE

       X1=POINT(1)*FACTOR(1)
       IF(X1.GT.TTTF.AND.(X1-TTTF).LE.1.)X1=X1+1.
       IF(X1.LE.TTTF.AND.(TTTF-X1).LE.1.)X1=X1-1.
       IBUF(KOUNT)=X1
       KOUNT=KOUNT+1
       Y1=POINT(2)*FACTOR(2)
       IF(Y1.GT.TTTF.AND.(Y1-TTTF).LE.1.)Y1=Y1+1.
       IF(Y1.LE.TTTF.AND.(TTTF-Y1).LE.1.)Y1=Y1-1.
       IBUF(KOUNT)=Y1
       KOUNT=KOUNT+1

C      GO TO (63,64,65),IFID
C63      IFID=2
C      POINT(1)=XCORZR
C      POINT(2)=YCORZR
C      IF(.NOT.PLOUT)GO TO 59
C      XX=XCORZR*PXFACT-.07
C      YY=YCORZR*PYFACT-.07
C      CALL SYMBOL(XX,YY,.14,"+",0.,1)
C      GO TO 59
C64      POINT(1)=P1
C      POINT(2)=P2
C      GO TO 50
C65      CONTINUE

                         ; IF A DASHED LINE, SEND TO CONDEN
      LPEN=IPEN
      IF(KPEN.EQ.5)LPEN=2
      IF(ITYPE.GT.1)CALL CONDEN(X,Y,LPEN,IBUF(KSAV),NCURV,ITYPE)

      IF(KOUNT.LT.LRECL)RETURN                  ; BUFFER NOT FULL YET

70       KOUNT=KOUNT-1

C        IF((.NOT.ABBREV).OR.(LTYPE.LE.2))WRITE(LCH,71)IBUF
         WRITE(LCH,71)IBUF
71       FORMAT(1X,10I7)
C       CALL MTDIO(LCH,052000K,IBUF,ISTAT,IER)
                              ; BUFFER FULL OR END OF CURVE:
```

```
                                    ; WRITE TO DISK
C        IF(IER.NE.1)TYPE " WRITE ERROR=",IER
        LPEN=IPEN
        IF(KPEN.EQ.10)LPEN=KPEN
                            ; IF END OF CURVE, SEND EOC FLAG TO CONDEN
        IF((LTYPE.GT.1).AND.(IBUF(KOUNT).EQ.177777K))
       1              CALL CONDEN(X,Y,LPEN,IBUF(KSAV),LASTC,LTYPE)

        DO 20 J=1,LRECL       ; ZERO BUFFER
20        IBUF(J)=0

        KOUNT=1
        IF(.NOT.HEAD)RETURN                      ; GET NEXT POINT

80        HEAD=.FALSE.
        INITL=.FALSE.
        LASTC=NCURV
        LTYPE=ITYPE
        KOUNT=1                     ; CURVE ENDED; SET UP HEADER

        IBUF(KOUNT)=NCURV                    ; CURVE NUMBER
        KOUNT=KOUNT+1
        IBUF(KOUNT)=0               ; UNABBREVIATED VERSION FLAG
        KOUNT=KOUNT+1
        IBUF(KOUNT)=ITYPE                    ; CURVE TYPE
        KOUNT=KOUNT+1
          IBUF(KOUNT)=0
          KOUNT=KOUNT+1
          X1=X*FACTOR(1)
          Y1=Y*FACTOR(2)
          IBUF(KOUNT)=IFIX(X1/TTTF+.001)
          IXSHFT=IBUF(KOUNT)
          KOUNT=KOUNT+1
          IBUF(KOUNT)=IFIX(Y1/TTTF+.001)
          IYSHFT=IBUF(KOUNT)
          KOUNT=KOUNT+1

          GO TO 50


C        IBUF(KOUNT)=2                           ; 2 PAIRS OF X FIDS
C        KOUNT=KOUNT+1
C        P1=POINT(1)
C        P2=POINT(2)
C        POINT(1)=XCORZL
C        POINT(2)=YCORZL
C        MBIT=0
C        IFID=1
C        IF(.NOT.PLOUT)GO TO 59
C        XX=XCORZL*PXFACT-.07
C        YY=YCORZL*PYFACT-.07
C        CALL SYMBOL(XX,YY,.14,"+",0.,1)
C        GO TO 59

         END

*)
```

```
      COMMON/SYSOFF/XOFF,YOFF,XFSCL,YFSCL,THAOFF,COSTHA,SINTHA          MTCH0001
      DIMENSION INFIL(4),INFDU(4)                                       MTCH0002
C                                                                       MTCH0003
C           THIS PROGRAM READS DIFFERENT SECTIONS OF 7-IN.              MTCH0004
C     RECORD CREATED BY KSCRIBE AND MATCHES THEM TOGETHER.              MTCH0005
C     AT MOST 10 SECTIONS ARE ALLOWED.                                  MTCH0006
C                                                                       MTCH0007
      DIMENSION XFIDL(10),YFIDL(10),XFIDR(10),YFIDR(10)                 MTCH0008
      COMMON/CURVE/ISC(10),ICURV(99,10)                                 MTCH0009
      COMMON/INDATA/INBUF(1024),                                        MTCH0010
     1 SBUF(1024),BUF(1024),NBUF(1024),HBUF(64),GBUF(64)                MTCH0011
      LOGICAL ALL,SKIP,START,CHECK,SKPLOT,NOPLOT                        MTCH0012
      IWRT=0                                                            MTCH0013
      NSEG=0                                                            MTCH0014
      FACTX=50.                                                         MTCH0015
      FACTY=50.                                                         MTCH0016
      CALL OPEN(0,"PLOTSCL.DT",1,IER)                                   MTCH0017
      READ FREE (0)XOFF,YOFF,XFSCL,YFSCL,THAOFF,COSTHA,SINTHA           MTCH0018
      CALL CLOSE(0,IER)                                                 MTCH0019
      CALL OPEN(0,"VOX01.DT",3,IER)                                     MTCH0020
      CALL OPEN(1,"VOXX1.DT",1,IER)                                     MTCH0021
      TYPE "          ****************************"                     MTCH0022
      TYPE "          *     INPUT INFORMATION     *"                    MTCH0023
      TYPE "           ***************************"                     MTCH0024
      TTTF=32768.                                                       MTCH0025
      LRECL=1024                                                        MTCH0026
      AMPI=25400.                                                       MTCH0027
      WRITE(10,1)                                                       MTCH0028
1     FORMAT(" INPUT TOTAL NUMBER OF SECTIONS TO BE MATCHED: "Z)        MTCH0029
      READ FREE (11)NSECS                                               MTCH0030
      IF(NSECS.GT.0..AND.NSECS.LT.9)GO TO 3                             MTCH0031
      WRITE(10,2)NSECS                                                  MTCH0032
2     FORMAT(" INPUT SECTION NO. =",I3," IS OUTSIDE RANGE. STOP.")      MTCH0033
      STOP                                                              MTCH0034
3     CONTINUE                                                          MTCH0035
      WRITE(10,4)                                                       MTCH0036
4     FORMAT(" INPUT TOTAL NO.OF SEGMENTS (SAME IN ALL SECTIONS): "Z)   MTCH0037
      READ FREE (11) NSEGS                                              MTCH0038
      WRITE(10,5)NSEGS,NSECS                                            MTCH0039
5     FORMAT(" FOR THE ",I2,                                            MTCH0040
     1       " SEGMENTS, INPUT THEIR SEGMENT NUMBERS IN THE "           MTCH0041
     1       ,I2," SECTIONS:")                                          MTCH0042
      DO 10 ISEC=1,NSECS                                                MTCH0043
      WRITE(10,6)ISEC,NSEGS                                             MTCH0044
6     FORMAT(" SECTION ",I2,"; ",I2,"  SEGMENT NUMBERS FROM TOP",       MTCH0045
     1 " TO BOTTOM ARE  (FREE FORMAT):")                                MTCH0046
      READ FREE (11)(ICURV(I,ISEC),I=1,NSEGS)                           MTCH0047
10    CONTINUE                                                          MTCH0048
11    FORMAT(10F7.0)                                                    MTCH0049
      WRITE(10,12)                                                      MTCH0050
12    FORMAT(" ALL CURVES TO BE PROCESSED (Y/N): "Z)                    MTCH0051
      READ (11,13)NSWER                                                 MTCH0052
13    FORMAT(A2)                                                        MTCH0053
      ALL=.TRUE.                                                        MTCH0054
      IF(NSWER.EQ."N ")ALL=.FALSE.                                      MTCH0055
      WRITE(10,457)                                                     MTCH0056
457   FORMAT(" INPUT HALFUP = "Z)                                       MTCH0057
      READ (11,459)HALFUP                                               MTCH0058
459   FORMAT(F10.3)                                                     MTCH0059
      CALL INITIAL(6,200,11.,50.)                                       MTCH0060
      DO 1000 ISEG=1,99                                                 MTCH0061
      SKIP=.FALSE.                                                      MTCH0062
      START=.FALSE.                                                     MTCH0063
      CHECK=.FALSE.                                                     MTCH0064
      SKPLOT=.FALSE.                                                    MTCH0065
      NOPLOT=.FALSE.                                                    MTCH0066
```

```
        IUNIT=1
        LCURV=0
        ISEC=1

        XLAST=0.
        XSAVE=0.
        XCORZN=0.
        YCORZN=0.
        NJ=0
        IST=7
21      CONTINUE
        READ(IUNIT,11,END=1001)(BUF(J),J=1,LRECL)
                        ; READ A BLOCK OF SEGMENT DATA
        DO 23 J=1,LRECL
        IF(BUF(J).EQ.TTTF)BUF(J)=BUF(J)-1.
        IF(BUF(J).EQ.-TTTF)BUF(J)=BUF(J)+1.
23      INBUF(J)=BUF(J)
C
31      CONTINUE
        IF(IST.EQ.1)GO TO 50
C
C       DECODE
C

        KCURV=INBUF(1)
        KFLAG=INBUF(2)
        KTYPE=INBUF(3)
C       CALL RSTR(0)
        IF(ISEC.GT.1)WRITE(10,38)KCURV,KTYPE,ISEC
38      FORMAT(20X,"DOING SEGMENT NO.",I4,", TYPE",I2,", IN SECTION",I2)

        IF(KTYPE.EQ.2.AND.KFLAG.EQ.1)TYPE " ABBREVIATED TIME VERSION."
        IF(KTYPE.EQ.3)SKIP=.TRUE.
        IF(KTYPE.EQ.2.AND.KFLAG.EQ.1)SKPLOT=.TRUE.
        IF(KTYPE.EQ.2.AND.KFLAG.EQ.0)NOPLOT=.TRUE.
        IXSH=INBUF(5)
        IYSH=INBUF(6)
        X0=INBUF(7)
        Y0=INBUF(8)
        XSHIFT=0.
        YSHIFT=0.
41      X1=X0+XSHIFT
        IX1=IFIX(X1/TTTF+.001)
        IF(IX1.EQ.IXSH)GO TO 43
        XSHIFT=XSHIFT+TTTF
        GO TO 41
43      Y1=Y0+YSHIFT
        IY1=IFIX(Y1/TTTF+.001)
        IF(IY1.EQ.IYSH)GO TO 45
        YSHIFT=YSHIFT+TTTF
        GO TO 43
45      CONTINUE
C

        X0=-TTTF
        NSKIP=2
        IF(ISEC.GT.1)GO TO 50
C
C               FIRST SECTION ONLY
C
C

        DO 40 J=1,6
40      SBUF(J)=BUF(J)
        NJ=6
CC
C               GET KSEG
        KSEG=0
32      KSEG=KSEG+1
```

```
      IF(KSEG.LE.NSEGS)GO TO 33
C     CALL RSTR(0)
      WRITE(10,325)KCURV
325   FORMAT(" SEGMENT #",I3," IN SECTION 1 IS BEING SKIPPED.")
      SKIP=.TRUE.
33    CONTINUE
      IF(NSEG.GE.NSEGS)GO TO 1001
      IF(SKIP)GO TO 50
      LCURV=ICURV(KSEG,1)
      IF(LCURV.NE.KCURV)GO TO 32
      IF(ALL)GO TO 35
C     CALL RSTR(0)
      WRITE(10,34)LCURV
34    FORMAT(" CURVE #",I3," TO BE PROCESSED (Y/N): "Z)
      READ (11,13)NSWER
      SKIP=.FALSE.
      IF(NSWER.EQ."N ")SKIP=.TRUE.
      IF(SKIP)GO TO 50
35    CONTINUE
      IF(KTYPE.GE.2.AND.KFLAG.EQ.1)NSEG=NSEG+1
      IF(KTYPE.LT.2)NSEG=NSEG+1
      IF(NSEG.GT.NSEGS)GO TO 1001
C
C
C     TYPE "****************************************"
C     TYPE "* START NEW SEGMENT FROM FIRST SECTION *"
C     TYPE "****************************************"
C
C
C     CALL RSTR(0)
      TYPE " *****************"
      TYPE "  CURVE",KCURV
      TYPE " *****************"
      WRITE(10,38)KCURV,KTYPE,ISEC
C
C             PLOT THE CURVE NUMBER
C
      PX=X1/AMPI
      PY=Y1/AMPI
      CURV=KCURV
      IF(.NOT.NOPLOT)CALL SYSNUM(PX-.3,PY-.07,.14,CURV,0.,-1)
      IPEN=3
C
C             GET THE CLOSEST FIDUCIAL PAIRS
      CALL OPEN(2,"VOFD1.DT",1,IER)
C     CALL RSTR(0)
      WRITE(10,455)X1,Y1
455   FORMAT(" FIRST POINT ON CURVE: X1,Y1 = ",2F10.2)
C             HALFUP PREVIOUSLY SET BY SCRIBE TO BE 5.5
C             PYFACT = DENSITOMETER UNITS/INCH
      PYFACT=AMPI/FACTY
      READ(2,46)NFIDS
      IF(IWRT.EQ.1)TYPE " NUMBER OF FIDUCIALS =",NFIDS
46    FORMAT(3X,I3)
      KFID=0
      DMIN=999999.
      DO 48 N=1,NFIDS
      READ(2,47)XL,YL,XR,YR
      IF(N.GT.1)GO TO 471
      YNORM=HALFUP*PYFACT-YL
      YL=YL+YNORM
      YDIF1=YL*FACTY
471   CONTINUE
      IF(N.GT.1)YL=YL+YNORM
      YR=YR+YNORM
47    FORMAT(4F10.1)
```

```
      DIST=SQRT((X1-XL*FACTX)**2+(Y1-YL*FACTY)**2)
      IF(IWRT.EQ.1)WRITE(10,47)XL,YL,XR,YR,DIST
      IF(DIST.GT.DMIN)GO TO 48
      DMIN=DIST
      KFID=N
      XFIDL(1)=XL*FACTX
      YFIDL(1)=YL*FACTY
      XFIDR(1)=(XL+SQRT((XR-XL)**2+(YR-YL)**2))*FACTX
      YFIDR(1)=YL*FACTY
48    CONTINUE
      YDIF1=YDIF1-YFIDL(1)
      CALL CLOSE(2,IER)
      TYPE " FIDUCIAL   #",KFID," TO BE USED."
      IF(KFID.GT.0)GO TO 50
      WRITE(10,49)
49    FORMAT(" FIDUCIAL PAIRS FOR THE TRACE NOT FOUND. CHECK.")
      STOP
50    CONTINUE
      DO 60 J=IST,LRECL,NSKIP
      IF(INBUF(J).EQ.177777K.AND.INBUF(J+1).EQ.177777K)GO TO 63
                      ; END OF CURVE
      IF(SKIP)GO TO 60
      X=INBUF(J)
      Y=INBUF(J+1)
      IF(X.GT.X0)GO TO 54
      IF(X.LT.0.)XSHIFT=XSHIFT+2.*TTTF
      IF(X.GE.0.)XSHIFT=XSHIFT+TTTF
54    CONTINUE
      X0=X
      Y0=Y
      X=X+XSHIFT+XCORZN
      Y=Y+YSHIFT+YCORZN
      IF(.NOT.CHECK)GO TO 57
C     IF(X.LT.XSAVE.OR.X.GT.XLAST)GO TO 57
      IF(X.LT.XSAVE)GO TO 57
      IF(X.GT.XLAST)GO TO 547
      IF(.NOT.START)GO TO 545
C     CALL RSTR(0)
      TYPE " START CHECKING."
      NC=3
541   IF(X.LT.HBUF(NC))GO TO 543
      NC=NC+2
      IF(NC.LE.NPOINT)GO TO 541
      WRITE(10,542)LSEC,ISEC
542   FORMAT(" GAP FOUND BETWEEN SECTION",I3," AND SECTION",I3,
     1 " NO MATCHUP.")
543   CONTINUE
      START=.FALSE.
      NB=NC-3
      ND=NB+1
      TYPE " FIRST POINT OVERLAPPED =",X,HBUF(ND)
545   CONTINUE
      NB=NB+2
      IF(NB.GE.NPOINT)GO TO 547
      NS=NS+1
      GBUF(NS)=X
      NS=NS+1
      GBUF(NS)=Y
      NR=NR+1
      ERRSQ=ERRSQ+(HBUF(NB)-Y)**2
      ERRAV=SQRT(ERRSQ/FLOAT(NR))
      TYPE " HBUF(NB),Y =",HBUF(NB),Y
      TYPE " ERRSQ,ERRAV =",ERRSQ,ERRAV
      GO TO 57
547   CONTINUE
      TYPE "X,XLAST =",X,XLAST
```

```
      WRITE(10,55)ERRAV,ERRMX
55    FORMAT("SAMPLE VARIANCE OF DEVIATION =",F10.5,"     MAX. = "
     1 F15.6)
      IF(SKPLOT)GO TO 56
      IF(NR.EQ.0)WRITE(10,542)LSEC,ISEC
      IF(NR.GT.0)
     1 CALL MCHECK(NR,HBUF(ND),GBUF,FACTX,FACTY,XCORZN,YCORZN,ERRAV)
      IF(ERRAV.LE.ERRMX)GO TO 56
      WRITE(10,55)ERRAV,ERRMX
      STOP
56    TYPE " MATCHUP !"
      CHECK=.FALSE.
57    CONTINUE
C     IF(CHECK)WRITE(10,55)ERRAV,ERRMX
      IF(NOPLOT)GO TO 58
C
C             PLOT THE POINT
      PX=X/AMPI
      PY=Y/AMPI
      CALL SYSPLT(PX,PY,IPEN)
      IF(.NOT.SKPLOT)IPEN=2
      IF(SKPLOT)IPEN=5-IPEN
58    CONTINUE
      IF(X.LE.XLAST)GO TO 60
C
C
C
C             STORE THE POINTS
      NJ=NJ+1
      SBUF(NJ)=X
      NJ=NJ+1
      SBUF(NJ)=Y
      IF(NJ.LT.LRECL)GO TO 60
C     IF(SKPLOT)CALL LCHECK(LRECL,SBUF)
      DO 59 JJJ=1,LRECL
59    NBUF(JJJ)=SBUF(JJJ)
      WRITE(0,62)(NBUF(JJJ),JJJ=1,LRECL)
      NJ=0
C
60    CONTINUE
61    FORMAT(10I7)
C62   FORMAT(1X,10I7)                  ;FOR RDOS FORTRAN IV
62    FORMAT(10I7)                  ;FOR RDOS FORTRAN V

C     IF(.NOT.CHECK)GO TO 623
C     IF(ERRAV.LE.ERRMX)GO TO 622
C     WRITE(10,55)ERRAV,ERRMX
C     STOP
C622  CHECK=.FALSE.
C623  CONTINUE
      IST=1
      GO TO 21
63    CONTINUE
      IF(SKIP)GO TO 1000
C
C             REMEMBER LAST 32 POINTS
C
      IF(IWRT.EQ.1)TYPE " LAST PORTION OF DATA READ IN."
      XLAST=X
C     NPOINT=MIN0(NJ/2,64)
      NPOINT=64
      IF(SKPLOT)NPOINT=MIN0(NJ/2,64)
      JJ=NJ
      DO 64 J=NPOINT,1,-1
      HBUF(J)=SBUF(JJ)
      JJ=JJ-1
```

```
      IF(JJ.GT.0)GO TO 64                                        MTCH0330
      JJ=LRECL                                                   MTCH0331
64    CONTINUE                                                   MTCH0332
      IF(IWRT.EQ.1)WRITE FREE (10) (HBUF(J),J=1,NPOINT)          MTCH0333
      XSAVE=HBUF(1)                                              MTCH0334
      IPEN=3                                                     MTCH0335
      ERRSQ=0.                                                   MTCH0336
      NB=0                                                       MTCH0337
C     TYPE " NPOINT =",NPOINT                                    MTCH0338
      ERRMX=FACTY*8.                                             MTCH0339
      NR=0                                                       MTCH0340
      NS=0                                                       MTCH0341
      CHECK=.TRUE.                                               MTCH0342
      START=.TRUE.                                               MTCH0343
C                                                                MTCH0344
C             GO TO NEXT SECTION                                 MTCH0345
C                                                                MTCH0346
      IF(IUNIT.EQ.2)CALL CLOSE(2,IER)                            MTCH0347
      ISEC=ISEC+1                                                MTCH0348
      IF(ISEC.GT.NSECS)GO TO 900                                 MTCH0349
      CALL OPEN(2,"INFILES.DT",1,IER)                            MTCH0350
      DO 643 JS=1,ISEC                                           MTCH0351
      READ(2,647)INFDU                                           MTCH0352
643   READ(2,647)                                                MTCH0353
      CALL CLOSE(2,IER)                                          MTCH0354
647   FORMAT(1X,4S2)                                             MTCH0355
C                                                                MTCH0356
C     GET CURVE NUMBER OF CONTINUING CURVE IN NEXT SECTION       MTCH0357
      KCURV=ICURV(KSEG,ISEC)                                     MTCH0358
      IF(IWRT.EQ.1)TYPE " CURVE NO.",KCURV,"  SECTION",ISEC      MTCH0359
C                                                                MTCH0360
C                                                                MTCH0361
C     GET FIDUCAIL PAIRS                                         MTCH0362
C                                                                MTCH0363
      CALL OPEN(2,INFDU,1,IER)                                   MTCH0364
      READ(2,46)KFIDS                                            MTCH0365
      IF(KFIDS.EQ.NFIDS)GO TO 67                                 MTCH0366
      WRITE(10,65)NFIDS,KFIDS,ISEC                               MTCH0367
65    FORMAT(" NO. OF FIDUCIALS IN SECTION 1,"I3," .NE. THAT IN" MTCH0368
     1" SECTION",I2,", CHECK.")                                  MTCH0369
      STOP                                                       MTCH0370
67    CONTINUE                                                   MTCH0371
      IF(IWRT.EQ.1)TYPE " KFID USED FOR NEXT SECTION =",KFID     MTCH0372
      READ(2,47)XL,YL,XR,YR                                      MTCH0373
      YNORM=HALFUP*PYFACT-YL                                     MTCH0374
      YL=YL+YNORM                                                MTCH0375
      YDIF2=FACTY*YL                                             MTCH0376
      IF(KFID.EQ.1)GO TO 691                                     MTCH0377
      DO 69 K=2,KFID                                             MTCH0378
      READ(2,47)XL,YL,XR,YR                                      MTCH0379
      YL=YL+YNORM                                                MTCH0380
69    CONTINUE                                                   MTCH0381
691   CONTINUE                                                   MTCH0382
      YR=YR+YNORM                                                MTCH0383
      XFIDL(ISEC)=XL*FACTX                                       MTCH0384
      YFIDL(ISEC)=YL*FACTY                                       MTCH0385
      XFIDR(ISEC)=(XL+SQRT((XR-XL)**2+(YR-YL)**2))*FACTX         MTCH0386
      YFIDR(ISEC)=YL*FACTY                                       MTCH0387
      YDIF2=YDIF2-YFIDL(ISEC)                                    MTCH0388
      CALL CLOSE(2,IER)                                          MTCH0389
      LSEC=ISEC-1                                                MTCH0390
C     XCORZN=XFIDR(LSEC)-XFIDL(ISEC)                             MTCH0391
C     YCORZN=YFIDR(LSEC)-YFIDL(ISEC)                             MTCH0392
      XCORZN=XCORZN+XFIDR(LSEC)-XFIDL(ISEC)                      MTCH0393
      YCORZN=0.                                                  MTCH0394
C     YCORZN=YDIF2-YDIF1                                         MTCH0395
```

```
      IF(IWRT.EQ.1)WRITE(10,695)XCORZN,YCORZN
695   FORMAT(" XCORZN,YCORZN =",2F12.3)
C
      CALL OPEN(2,"INFILES.DT",1,IER)
      DO 697 JS=1,ISEC
      READ(2,647)
697   READ(2,647)INFIL
      CALL CLOSE(2,IER)
C
C
C     CALL OPEN(2,"VOXX2.DT",1,IER)
      CALL OPEN(2,INFIL,1,IER)
      IF(IWRT.EQ.1)TYPE "IER =",IER
70    CONTINUE
      IUNIT=2
      IST=7
      IF(IWRT.EQ.1)TYPE " READING BEGINNING OF NEXT SECTION:",ISEC
71    CONTINUE
      READ(2,11,END=81)(BUF(J),J=1,LRECL)
C     TYPE " 1 BLOCK OF DATA READ."
      DO 72 J=1,LRECL
72    INBUF(J)=BUF(J)
      IF(IST.EQ.1)GO TO 73
      JCURV=INBUF(1)
      JFLAG=INBUF(2)
      JTYPE=INBUF(3)
      IF(IWRT.EQ.1)
     1 TYPE " JCURV =",JCURV," JFLAG =",JFLAG," JTYPE=",JTYPE
      IF(JCURV.EQ.KCURV.AND.JFLAG.EQ.KFLAG)GO TO 76
73    CONTINUE
C     SKIP TO NEXT BLOCK OF DATA
      DO 75 J=IST,LRECL,NSKIP
      IF(INBUF(J).EQ.177777K.AND.INBUF(J+1).EQ.177777K)GO TO 70
75    CONTINUE
      IST=1
      GO TO 71
76    CONTINUE
      IF(JTYPE.EQ.KTYPE)GO TO 78
      WRITE(10,77)KCURV,KTYPE,JCURV,JTYPE,ISEC
77    FORMAT(" CURVE NO.",I4,", OF TYPE",I2," IS ASSOCIATED WITH"/10X
     1 ,"CURVE NO.",I4,", OF TYPE",I2," IN SECTION",I3,". WRONG TYPE.")
      CALL CLOSE(2,IER)
      STOP
78    CONTINUE
      DO 79 J=1,LRECL
      IF(BUF(J).EQ.TTTF)BUF(J)=BUF(J)-1.
      IF(BUF(J).EQ.-TTTF)BUF(J)=BUF(J)+1.
79    INBUF(J)=BUF(J)
C
C     CALL RSTR(0)
      TYPE " CHECK FOR MATCHING"
C
      GO TO 31
81    CONTINUE
      WRITE(10,83)KCURV,ISEC
83    FORMAT(" CURVE NO.",I3," INPUT FROM TERMINAL NOT FOUND IN"
     1 ," SECTION",I3)
      STOP
900   CONTINUE
      NJ=NJ+1
      SBUF(NJ)=-1.01
      NJ=NJ+1
      SBUF(NJ)=-1.01
      IF(NJ.EQ.LRECL)GO TO 920
      NJ1=NJ+1
      DO 910 JJJ=NJ1,LRECL
```

```
910     SBUF(JJJ)=0.                                              MTCH0462
920     CONTINUE                                                  MTCH0463
        IF(SKPLOT)CALL LCHECK(LRECL,SBUF)                         MTCH0464
        DO 930 JJJ=1,LRECL                                        MTCH0465
930     NBUF(JJJ)=SBUF(JJJ)                                       MTCH0466
        WRITE(0,62)(NBUF(JJJ),JJJ=1,LRECL)                        MTCH0467
        NJ=0                                                      MTCH0468
C       CALL RSTR(0)                                              MTCH0469
1000    CONTINUE                                                  MTCH0470
1001    CONTINUE                                                  MTCH0471
        CALL CLOSE(0,IER)                                         MTCH0472
        CALL CLOSE(1,IER)                                         MTCH0473
C       CALL RSTR(2)                                              MTCH0474
        STOP                                                      MTCH0475
        END                                                       MTCH0476
```

```
      SUBROUTINE MCHECK                                                 MCHK0001
     1 (NBUF,HBUF,GBUF,FACTX,FACTY,XCORZN,YCORZN,ERRMIN)                MCHK0002
      DIMENSION HBUF(1),GBUF(1),HNEW(100),GNEW(100),ERRAV(-5:5,-5:5)    MCHK0003
      IWRT=13                                                           MCHK0004
      CALL FOPEN(IWRT,"SCRTCH.DT")                                      MCHK0005
      WRITE(IWRT,1)(HBUF(I),I=1,NBUF)                                   MCHK0006
      WRITE(IWRT,1)(GBUF(I),I=1,NBUF)                                   MCHK0007
      CALL INTRPL(NBUF,HBUF,NEW1,HNEW)                                  MCHK0008
      CALL INTRPL(NBUF,GBUF,NEW2,GNEW)                                  MCHK0009
      NNEW=MIN0(NEW1,NEW2)                                              MCHK0010
      WRITE(IWRT,1)(HNEW(I),I=1,NNEW)                                   MCHK0011
      WRITE(IWRT,1)(GNEW(I),I=1,NNEW)                                   MCHK0012
1     FORMAT(8F10.2)                                                    MCHK0013
      NLEFT=-4                                                          MCHK0014
      NRITE=4                                                           MCHK0015
      DO 60 N=NLEFT,NRITE                                               MCHK0016
      YT=FLOAT(N)*FACTY                                                 MCHK0017
C     TYPE " N,YT =",N,YT                                               MCHK0018
      DO 50 M=NLEFT,NRITE                                               MCHK0019
      IF(M)20,30,30                                                     MCHK0020
20    ISTART=2*(-M+1)                                                   MCHK0021
      JJJ=0                                                             MCHK0022
      ERRSQ=0.                                                          MCHK0023
      DO 21 III=ISTART,NNEW,2                                           MCHK0024
      JJJ=JJJ+1                                                         MCHK0025
      GY=GNEW(III)+YT                                                   MCHK0026
      HY=HNEW(2*JJJ)                                                    MCHK0027
      ERRSQ=ERRSQ+(GY-HY)**2                                            MCHK0028
21    CONTINUE                                                          MCHK0029
      NR=JJJ                                                            MCHK0030
      GO TO 40                                                          MCHK0031
30    JSTART=2*(M+1)                                                    MCHK0032
      III=0                                                             MCHK0033
      ERRSQ=0.                                                          MCHK0034
      DO 31 JJJ=JSTART,NNEW,2                                           MCHK0035
      III=III+1                                                         MCHK0036
      GY=GNEW(2*III)+YT                                                 MCHK0037
      HY=HNEW(JJJ)                                                      MCHK0038
      ERRSQ=ERRSQ+(GY-HY)**2                                            MCHK0039
      CONTINUE                                                          MCHK0040
      NR=III                                                            MCHK0041
```

```
40          CONTINUE
            ERRAV(M,N)=SQRT(ERRSQ/FLOAT(NR))
50          CONTINUE
            WRITE(IWRT,51)(ERRAV(M,N),M=NLEFT,NRITE)
C           WRITE(10,51)(ERRAV(M,N),M=NLEFT,NRITE)
51          FORMAT(8F10.2)
60          CONTINUE
C
C           FIND THE BEST TRANSLATION
            IMIN=NLEFT
            JMIN=NLEFT
            ERRMIN=ERRAV(IMIN,JMIN)
            DO 70 J=NLEFT,NRITE
            DO 70 I=NLEFT,NRITE
            IF(ERRAV(I,J).GE.ERRMIN)GO TO 70
            ERRMIN=ERRAV(I,J)
            IMIN=I
            JMIN=J
70          CONTINUE
            WRITE(IWRT,71)XCORZN,YCORZN
C           WRITE(10,71)XCORZN,YCORZN
71          FORMAT(" BEFORE TRANSLATION, XCORZN,YCORZN = ",2F10.2)
            XDIF=HBUF(1)-GBUF(1)
            WRITE(IWRT,715)IMIN,JMIN,ERRMIN
715         FORMAT(" IMIN,JMIN,ERRMIN = ",2I5,F10.2)
            TYPE " IMIN,JMIN,ERRMIN =",IMIN,JMIN,ERRMIN
            XCORZN=XCORZN+XDIF+FLOAT(IMIN)*FACTX
            YCORZN=YCORZN+FLOAT(JMIN)*FACTY
            WRITE(IWRT,72)XCORZN,YCORZN
C           WRITE(10,72)XCORZN,YCORZN
72          FORMAT(" AFTER TRANSLATION, XCORZN,YCORZN = ",2F10.2)
            CALL FCLOSE(IWRT)
            RETURN
            END
```

```
            SUBROUTINE INTRPL(NBUF,HBUF,NNEW,HNEW)
            DIMENSION HBUF(1),HNEW(1)
            FACT=50.
            FACT1=FACT+1.
            NMAX=100
            HNEW(1)=HBUF(1)
            HNEW(2)=HBUF(2)
            I=1
            J=1
11          CONTINUE
            I0=I
            J0=J
            I=I+2
            IF(I.GT.NBUF)GO TO 99
            J=J+2
            IF(J.GT.NMAX)GO TO 99
            I1=I+1
            J1=J+1
            HDIF=HBUF(I)-HBUF(I0)
            IF(HDIF.GT.FACT1)GO TO 13
            HNEW(J)=HBUF(I)
            HNEW(J1)=HBUF(I1)
```

```
      NNEW=J1
      GO TO 11
13    CONTINUE
      JAS=IFIX(HDIF/FACT+.1)
      HSLP=(HBUF(I1)-HBUF(I0+1))/FLOAT(JAS)
      DO 15 JA=1,JAS
      J1=J+1
      HNEW(J)=HNEW(J0)+FACT
      HNEW(J1)=HBUF(I0+1)+HSLP*FLOAT(JA)
      NNEW=J1
      J0=J
      J=J+2
      IF(J.GT.NMAX)GO TO 99
15    CONTINUE
      J=J-2
      GO TO 11
99    RETURN
      END
```

```
      SUBROUTINE LCHECK(KRECL,SBUF)
      DIMENSION SBUF(1)
      CALL FOPEN(5,"SCRTCH.DT")
C     CALL OPEN(5,"SCRTCH.DT",3,IER)
      LRECL=KRECL
      LRECL2=LRECL-2
      EPSI=.1
      WRITE(5,11)(SBUF(I),I=1,LRECL)
11    FORMAT(8F10.0)
      ICOUNT=0
      DO 3 JJ=1,LRECL,2
      IF(ABS(SBUF(JJ)+1.).LT.EPSI.AND.ABS(SBUF(JJ+1)+1.).LT.EPSI)
     1 GO TO 5
      ICOUNT=ICOUNT+2
3     CONTINUE
5     NJ=ICOUNT
12    CONTINUE
      NJ=NJ-2
      IERR=0
      DASHX=SBUF(13)-SBUF(11)
      ICOUNT=1
      DO 21 JJ=15,NJ,4
      DX=SBUF(JJ+2)-SBUF(JJ)
      RATIO=ABS(DX-DASHX)/DASHX
      IF(RATIO.GE.0.2)GO TO 15
      ICONT1=ICOUNT+1
      DASHX=(DASHX*FLOAT(ICOUNT)+DX)/FLOAT(ICONT1)
      ICOUNT=ICONT1
      GO TO 21
15    CONTINUE
      TYPE " ILLEGAL TIME MARKS AROUND X =",SBUF(JJ)
      IERR=1
      JS=JJ+2
      DO 17 JJJ=JS,LRECL2
17    SBUF(JJJ)=SBUF(JJJ+2)
      LRECL1=LRECL-1
      SBUF(LRECL1)=0.
      SBUF(LRECL)=0.
```

```
      GO TO 12                                                          LCHK0039
21    CONTINUE                                                          LCHK0040
31    CONTINUE                                                          LCHK0041
      WRITE(5,11)(SBUF(JJJ),JJJ=1,LRECL)                                LCHK0042
      CALL CLOSE(5,IER)                                                 LCHK0043
      RETURN                                                            LCHK0044
      END                                                               LCHK0045
```

APPENDIX B

THE UPDATED AUTOMATIC ROUTINE DIGITIZATION SYSTEM (ARDS) II

B.1 7-Inch Film Records

The ARDS System at USC was originally developed to digitize accelerograph records on 70 mm film. A typical strong motion acceleration may last for tens of seconds and, if the complete length of useful recording is considered, many digitized accelerograms approach the duration of one minute. In most cases, this is still shorter than 92 sec. which can be accommodated on one 10" by 10" negative used in automatic digitization. The film copying mechanism (Trifunac and Lee, 1979) which was specifically developed for ARDS converts four sequential 23 cm-sections of the original film onto a 10 x 10 in negative, with consecutive sections containing the corresponding fiducial marks. Short portions at the end of each 23 cm long section are repeated at the beginning of the following section to facilitate subsequent software reconstruction of the original record.

Recently, many buildings have been equipped with a centrally recording system which writes up to 13 simultaneous components of transducer outputs onto a film 7 inches in width. Digitization of one 10 x 10 in. negative copy from this type of instrument thus corresponds to that of only one 10-inch section of the 7-in. record with up to about 13 components. One such section corresponds to a duration of about 23 seconds. When the complete length of useful recording is desired, digitization of such accelerograms for a duration covering sequential 10 x 10 inch copies becomes necessary. A set of computer programs is thus necessary to handle digitization of more than one 10-inch consecutive sections.

When digitization and editing are completed for each section, "7SCRIBE" is called to plot and write the raw data onto disk files, a different file for each section of the record. "7SCRIBE" keeps track of the section number of each file and the total number of sections digitized, up to a maximum of 9.

Like "SCRIBE", "7SCRIBE" reads segments of each section, correcting for whatever fiducial skew may be present in that section. The fiducial coordinates of all the fiducial pairs of each section are also written onto disk, to prepare the data file for the next step. When all the data of each section digitized are written onto disk by "7SCRIBE", it is necessary to reconstruct a continuous curve of each segment out of all digitized sections. Different sections of the record now correspond to different files on disk and the program "MATCHUP" is used to connect them.

"MATCHUP" first asks the user for the total number of files present and for each file the curve identification numbers of all the segments present. Like "SCRIBE", "MATCHUP" will start reading through a segment in the first file. When the end of the segment in one file is reached, "MATCHUP" will identify the matching segment in the next file. Like "SCRIBE", "MATCHTUP" will then perform the necessary corrections - translations and rotations - to each coordinate of that segment in the next file. Each coordinate of the segment is also plotted out in the same scale as that of the original film. This continues until the last file is reached, thus reconstructing the continuous curve out of two or more files. This is done for each line present on the original film.

B.2 Sample Run: "7SCRIBE", "MATCHUP"

"7SCRIBE" is developed under the RDOS operating system for the NOVA 3 computer, the same operating system under which the whole process of digitization is carried out.

The running of "7SCRIBE" is similar to that of "KSCRIBE", with "7SCRIBE" specially designed for 7-in records. It allows for more than one set of fiducial pairs on each section. The user begins by typing:

```
) 7SCRIBE ↵
```

and "7SCRIBE" will respond with the message and question:

```
THIS IS A 7-IN. RECORD
INPUT SECTION NO. OF DATA (1,2,3,...)(I1):
```

The section no. keeps track of the sequential film number since each section now corresponds to a different digitization. The program allows up to a maximum of 9 sections. On reading in the section no. input by the user, N, say, where $1 \le N \le 9$, "7SCRIBE" will open a disk file VOXXN.DT for the Raw Data and VOFDN.DT for the fiducial coordinates of that section. The next 2 questions from "7SCRIBE" are same as those in "KSCRIBE":

```
ENTER MAPFILE SUFFIX:  MP
'P'LOT, 'D'ISK, OR 'B'OTH:
```

and the program will then proceed as "KSCRIBE".

Figures B.2-1 and B.2-2 are examples of plots of 2 files of Raw Data, VOXX1.DT and VOXX2.DT corresponding to 2 sections of a 7-in. record. The numbers at left identify each trace and are same as those selected by the program TRACE (Trifunac and Lee, 1979).

The next step, as an example, is to run "MATCHUP" to match-up the data in the 2 files VOXX1.DT and VOXX2.DT generated above by "7SCRIBE". The set of programs and subroutines which "MATCHUP" uses are developed under the AOS operating system for the Eclipse S/130 Time-sharing system. The user begins by typing:

```
)MATCHUP
```

and "MATCHUP" will respond with the message and question:

```
************************
*   INPUT INFORMATION   *
************************
INPUT TOTAL NUMBER OF SECTIONS TO BE MATCHED:
```

The user will type in, for this example (Figure B.2-1): 2. "MATCHUP" next asks:

```
INPUT TOTAL NUMBER OF SEGMENTS (SAME IN ALL SECTIONS):
```

and the user will type in, for this example: 20. "MATCHUP" then responds with

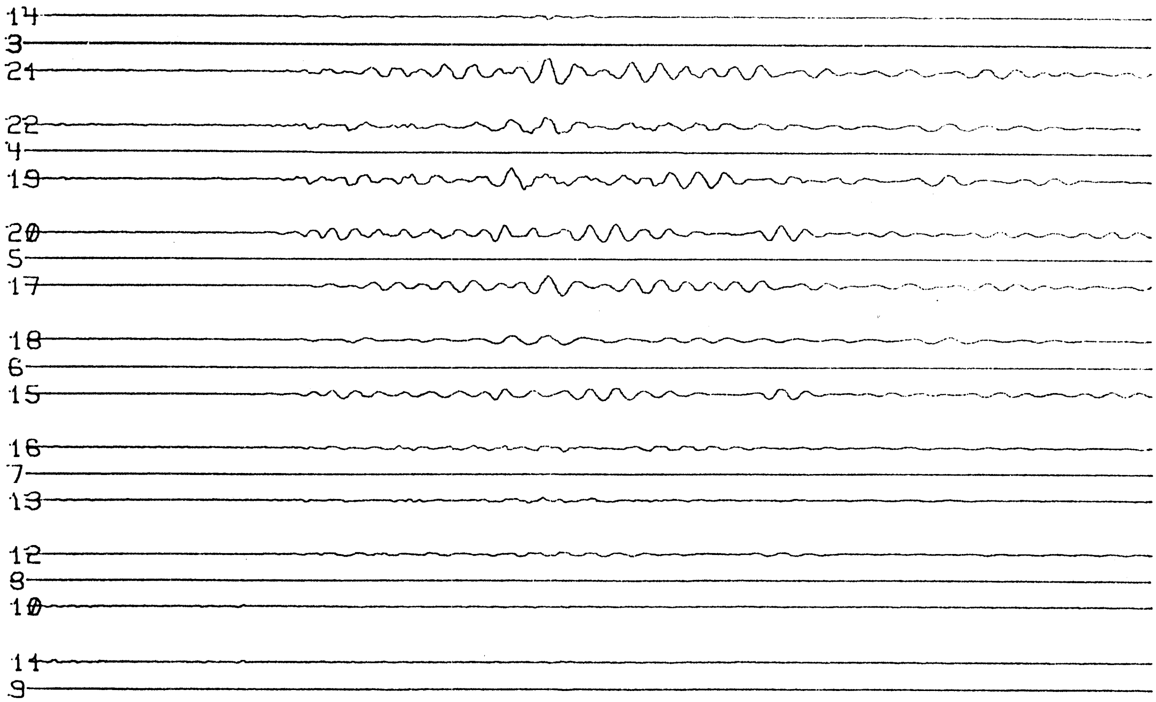


Figure B.2-1

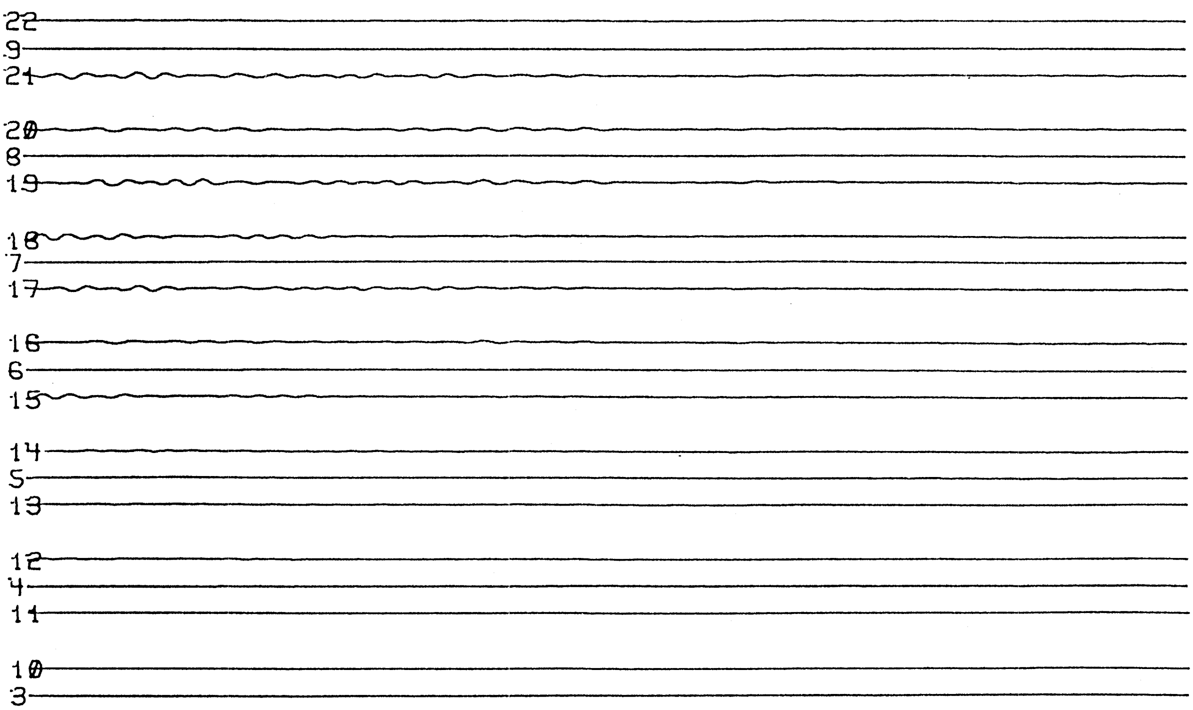


Figure B.2-2

```
FOR THE 20 SEGMENTS, INPUT THEIR SEGMENT NUMBERS
IN THE 2 SECTIONS:
SECTION 1; 20 SEGMENT NUMBERS FROM TOP TO BOTTOM
ARE (FREE FORMAT):
```

The user, referring to Figure B.2-1 will type in the segment numbers from top to bottom as:

```
14,3,21,22,4,19,20,5.17,18,6,15,16,7,13,12,8,10,11,9
```

"MATCHUP" next asks the same question for the next section:

```
SECTION 2;  20 SEGMENT NUMBERS FROM TOP TO BOTTOM
ARE (FREE FORMAT):
```

Referring to Figure B.2-2, the user types in

```
22,9,21,20,8,19,18,7,17,16,6,15,14,5,13,12,4,11,10,3
```

"MATCHUP" will ask the final question:

```
All curves to be processed (Y/N):
```

A Y answer is most appropriate. The program will then proceed immediately starting from reading curve 1 of the first file, plotting on the plotter and writing the data onto the file "VOX01.DT", continuing into the second file. This will repeat for every curve until all curves are processed. The following plot of the completed match-up data is shown in Figure B.2-3.

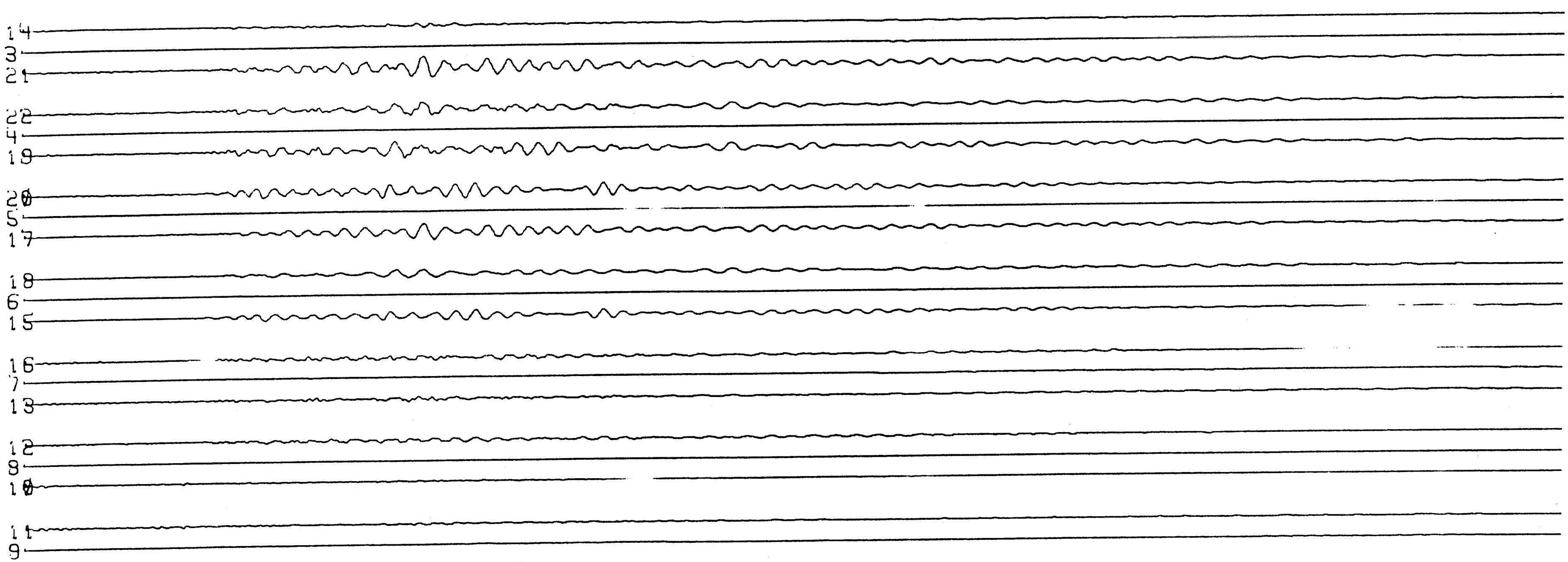


Figure B.2-3

```
 C      TITLE: REFCHK.F4
C
C  DETERMINES IF THE END OF A SEGMENT "POINTS" IN THE
C  GENERAL DIRECTION OF THE GIVEN PULSE.  A SEARCH BEAM
C  OF 90 DEGREES OR 30 DEGREES, THE FORMER WITH A SHORT
C  RADIUS, THE LATTER WITH A LONGER RADIUS, CAN BE USED.
C
C  ALSO CHECKS TO SEE IF A PULSE FALLS WITHIN NARROW
C  LIMITS FOR ASSOCIATION WITH A REFERENCE LINE.  IF
C  IT DOES, REFCHK LOOKS NO FURTHER.
C
C  CALLING SEQUENCE: CALL REFCHK(PULSE,KANDAT,ITRY)
C
C        WHERE PULSE CONTAINS THE PULSE'S PARAMETERS,
C              KANDAT IS ANY SEGMENT PASSING THE TEST
C              ITRY SIGNALS WHETHER TO USE WIDE SEARCH
C              OR NARROW.
C
C       FEB 20, 82        UPDATED TO INCLUDE RECTANGULAR SEARCH
C
C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C C

      PARAMETER POT=1.57   ;PI/2.
      PARAMETER POF=.7854   ;PI/4.=HALF THE WIDE SEARCH BEAM ANGLE
      PARAMETER POS=.2618   ;PI/12.=HALF THE NARROW SEARCH BEAM ANGLE
      PARAMETER SMALRAD=30.   ;SMALL SEARCH RADIUS
      PARAMETER BIGRAD=50.   ;BIG SEARCH RADIUS
      PARAMETER THIN=3.   ;MINIMUM WIDTH FOR NON-REF LINE
      PARAMETER GAP=4.   ;MAXIMUM DISTANCE IN X BETWEEN
                     ;CURRENT X AND SEG ENDPOINT.

      SUBROUTINE REFCHK(PULSE,KANDAT,ITRY)

        COMMON/LOCAL/DDX,DDY
      COMMON/CNSTNT/MAXSEG,IDUMMY(22)
      COMMON/TRACES/PARAM(20,75),SEGPTS(64,75)
      COMMON/ABSISA/X

      DIMENSION PULSE(2)

      Y=PULSE(2)+((PULSE(1)-1.)/2.)
      KANDAT=0
      RADIUS=BIGRAD   ;IF ITRY=2, USE LARGER SEARCH RADIUS,
      POX=POS         ;BUT THE NARROWER SEARCH BEAM.
      REFADD=1.   ;LOOSE REFERANCE-LINE TOLERENCE
      IF(ITRY.GT.1)GO TO 10

      POX=POF          ;WIDE SEARCH BEAM
      RADIUS=SMALRAD   ;SMALLER RADIUS
      REFADD=0.   ;TIGHT REF-LINE TOLERANCE

10      DO 1 J=1,MAXSEG   ;IF ITRY=1, FIRST CHECK REFERENCE LINES
      SEGEND=PARAM(1,J)
      IF(SEGEND.EQ.0.)GO TO 20
      STYPE=PARAM(9,J)
      IF(STYPE.EQ.0.)GO TO 1
      IF((X-SEGEND).LT.1.)GO TO 1
      YDIF=PARAM(10,J)-Y
      IF(ABS(YDIF).LT.(STYPE+REFADD))GO TO 2
1      CONTINUE

20      IF(PULSE(1).LT.THIN)RETURN

      DO 11 J=1,MAXSEG   ;NOW TRY ALL OTHER SEGS
      IF(PARAM(1,J).EQ.0.)GO TO 12
      IF(PARAM(9,J).NE.0.)GO TO 11
```

```
      IF(ABS(Y-PARAM(10,J)).GT.RADIUS)GO TO 11
      XDIF=X-PARAM(1,J)
      IF(XDIF.LT.1.)GO TO 11
      IF(XDIF.GT.GAP)GO TO 11
      IPPTS=IFIX(PARAM(4,J))
      IF(IPPTS.LT.2)GO TO 11
         ;PROJECT END OF SEGMENT TO PULSE'S X
         ;AND SEND OUT SEARCH BEAM.  IF BEAM EDGE
         ;BACKS UP IN X, ROTATE BEAM UNTIL
         ;IT DOESN'T.
      SY1=PARAM(14+IPPTS,J)
      SY2=PARAM(15+IPPTS,J)
      YDIF=SY2-SY1
      ROTAT=0.
      IF(ITRY.GT.1)GO TO 25
      HEDLYT=SY1
      XDIF=XDIF+(PARAM(12+IPPTS,J)-PARAM(11+IPPTS,J))
      GO TO 26

25       HEDLYT=SY2
26       ALPHA=ATAN2(YDIF,1.)
      THETAT=ALPHA+POX
      IF(THETAT.LE.POT)GO TO 21
      ROTAT=POT-THETAT

21       THETAB=ALPHA-POX
      IF(THETAB.GE.-POT)GO TO 22
      ROTAT=(-POT)-THETAB

22       THETAT=THETAT+ROTAT
      THETAB=THETAB+ROTAT
      YTOP=XDIF*TAN(THETAT) + HEDLYT
      IF(Y.GT.YTOP)GO TO 11
      YBOT=XDIF*TAN(THETAB) + HEDLYT
      IF(Y.LT.YBOT)GO TO 11
      IF(KANDAT.NE.0)GO TO 15
      KANDAT=J
11       CONTINUE

12       IF(KANDAT.NE.0)GO TO 1000

15       KANDAT=0
      GO TO 1000

2       KANDAT=J

1000     IF(KANDAT.NE.0)RETURN
         IF(ITRY.NE.2)RETURN
         ISTPS=3
         DO 102 ISTP=1,ISTPS
         YWIDTH=DDY*FLOAT(ISTP)
         DO 101 J=1,MAXSEG
         IF(PARAM(1,J).EQ.0.)GO TO 102
         IF(PARAM(9,J).GT.0.01)GO TO 101
                           ; CONSIDER ONLY NON-REFERENCE TYPE SEGMENT
         IF(ABS(Y-PARAM(10,J)).GT.YWIDTH)GO TO 101
         XDIF=X-PARAM(1,J)
         IF(XDIF.LT..95)GO TO 101
         IF(XDIF.GT.DDX)GO TO 101
         KANDAT=J
         GO TO 110
101      CONTINUE
102      CONTINUE
110      RETURN
      END
```