29 August 1975: se

Plot Program

by Gürcan Aral

In this study, we were mainly concerned with the development of a more or less self-contained facility for on-line collection, processing and display of seismic data. The facility used is the Information Systems Laboratory mini-computer. The computer is an INTERDATA 16 bit, 24k memory mini-computer that has as peripheral devices one teletype, one CRT-keyboard, one hardcopy unit from the CRT screen, a GOULD 5000 printer-plotter, a magnetic storage disc unit and a telephone line link to the Stanford Center for Information Processing (SCIP) IBM 360/67 computing system.

At present, it is possible to transfer data in character format from SCIP using the text editor WYLBUR at a rate of 300 BAUD which soon will be increased to a rate of 4800 BAUD. Since a WYLBUR file can accept up to approximately 9000 lines of 133 characters per line, about 1,000,000 characters can be stored in such a file. Typical seismic data consists of about 1500 samples per trace. Allowing a range of $\pm$ 999 per sample, it can be figured out that one trace takes about 6000 characters, permitting a WYLBUR file to accommodate about 160 traces. At the present rate of 30 characters per second, it takes about 10 hours to transfer one complete WYLBUR file onto the magnetic disc of the INTERDATA system. However, at a rate of 4800 BAUD, this time reduces to about 40 minutes. The 1,000,000 Bytes storage area on the magnetic disc is sufficient to store this data, in particular if numerical data is stored in its binary representation, rather than in character format.

The GOULD 5000 printer-plotter provides a high quality, high speed permanent output on 11 inch wide paper with a resolution of 100 grains per inch. Hardware character printing and software graphics display are possible.

For geophysical data display the following program, named "BEX" is available (Figure 1).

```
C---  B E X
C---  DISPLAYS SEISMIC DATA RECORDED ON CHANNEL 8 (DISC) IN INTEGER
C---  BINARY FORM ACCORDING TO THE PROVIDED AND DESIRED PARAMETERS
C---  PROVIDED PARAMETERS:
C---  SAMPRT:SAMPLE RATE IN MILLISECONDS/SAMPLE
C---  NOTR  :NUMBER OF TRACES RECORDED
C---  NSAMP :NUMBER OF SAMPLES/TRACE
C---  DESIRED PARAMETERS:
C---  TRSP  :TRACE SPACING IN (TRACES/INCH)
C---  CLIP  :DESIRED SATURATION VALUE
C---  VAP   :RATIO OF DARKENED TO THE TOTAL DISPLAY SURFACE IF NO
C---         SIGNAL IS PRESENT.
C---  TINCH :INCHES TO BE DISPLAYED ALONG A TRACE
C---  NSD   :NUMBER OF TRACES DESIRED TO BE DISPLAYED
C---
C---       RECORDS AND REFERENCES ARE USED.LINEPOINTER LNPTR(I)
C---       POINTS WITHIN A TRACE TO THE LAST COLUMN OF THE TRACE
C---       LINE I TO BE SET OR POINTS TO ZERO INDICATING THAT
C---       THERE ARE NO COLUMNS TO BE SET.THE LCOL(J) POINTS TO
C---       ITSELF INDICATING THE NEXT COLUMN OF LINE I TO BE SET
C---       AFTER COLUMN J OF LINE I IS SET.OR IT POINTS TO ZERO
C---       INDICATING THAT NO MORE COLUMNS AFTER J-TH COLUMN OF
C---       LINE I IS TO BE SET.HENCE WE CAN PROCEED TO THT
C---       NEXT LINE.
C-------------------------------------------------------------------
      INTEGER*2 CLIP
      TRSP=2
      TSC=20
      SAMPRT=1
      CLIP=30
      VAP=.5
      NOTR=1
      NSAMP=60
C-----USER DESIRED PARAMETERS
      TINCH=1
      NTD=1
C-----------------------------
      CALL DISP(TRSP,TSC,SAMPRT,CLIP,VAP,NOTR,NSAMP,TINCH,NTD)
      STOP
      END
C-------------------------------------------------------
```

Figure 1 continued on next page.

```
C
      SUBROUTINE DISP(TRSP,TSC,SAMPRT,CLIP,VAP,NOTR,NSAMP,TINCH,NTD)
      INTEGER*2 MASK(16),LINE(66),IDATA(1500),NDATA
      INTEGER*2 LCOL(1056),LNPTR(50),CLIP,INTER(1056)
      NS=NSAMP
      NT=NOTR
      NSLICE=100.49/TRSP+1.
      NSD=TINCH*1000/TSC/SAMPRT
      NDP=96*TINCH
      JFSET=-2*CLIP*VAP
      NDELTA=NS-NSD
      IND=NSLICE
      NTR=NT
      IF(NTD.LT.NT)NTR=NTD
      DO 45 N=1,16
      IF(N.EQ.1)MASK(N)=32767+1
45    IF(N.NE.1)MASK(N)=2**(16-N)
        DO 2 I=1,1056
2       IDATA(I)=0
      NMIN=NS
      IF(NSD.LT.NS)NMIN=NSD
      DO 65 I=1,NDP
65    INTER(I)=FLOAT(I-1)*NMIN/NDP+1
      WRITE(9,111)
111   FORMAT(3HBEX)
C---- LOOP FOR TRACES STARTS HERE
      DO 700 II=1,NTR
        DO 3 I=1,NMIN
        READ(8) NDATA
      IF(NDATA.GT.CLIP)NDATA=CLIP
      IF (NDATA.LT.JFSET)NDATA=JFSET
      IDATA(I)=(NDATA-JFSET)*(NSLICE-1)/(CLIP-JFSET)+1
3     CONTINUE
      IF (NS.LE.NSD)GO TO 4
C---- SKIP EXCESSIVE DATA
        DO 31 I=1,NDELTA
31      READ(8) NDATA
4           DO 20 I=1,66
20          LINE(I)=0
      DO 47 K=1,NSLICE
47    LNPTR(K)=0
      DO 46 I=1,NDP
      IJ=INTER(I)
      IVAL=IDATA(IJ)
      LCOL(I)=LNPTR(IVAL)
46    LNPTR(IVAL)=I
      MSLICE=NSLICE-1
      DO 100 M=1,MSLICE
      J=NSLICE-M+1
      IF(LNPTR(J).EQ.0) GO TO 100
      I=LNPTR(J)
40    IBYTE=1+(I-1)/16
      IBIT=1+MOD((I-1),16)
      LINE(IBYTE)=LOR(LINE(IBYTE),MASK(IBIT))
      I=LCOL(I)
      IF(I.NE.0) GO TO 40
100   CALL PRINT(LINE,1,9)
700   CONTINUE
      RETURN
      END
```

FIGURE 1.

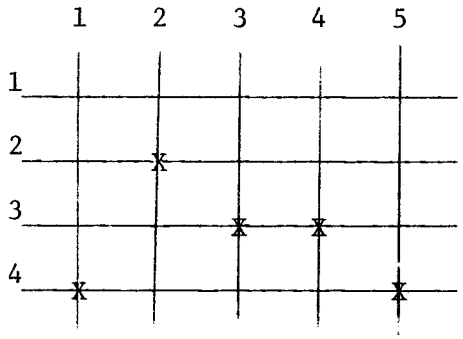
The program reads integer data in binary form, one data per card from the logical unit number 8 (disc) and displays it as requested by the initialized parameters on logical unit number 9 which is the printer-plotter.

Pointers used in the program perform as follows: LINEPOINTER (I) points to the last column of the I^{th} line to be set, while COLUMNPOINTER (J) points to the next column to be set. If either of them points to zero, it indicates that there are no more columns to be set, hence we may proceed to the next line.

Assume a data of

[4, 2, 3, 3, 4]

to be displayed as:



Then the pointers should contain:

line I	LINEPOINTER (I)
1	Ø (nothing to be set on line #1)
2	2 (last column to be set)
3	4 (" " " " ")
4	5 (" " " " ")

column J	COLUMNPOINTER (J)
1	∅
2	∅
3	∅
4	3
5	1

To display the information, proceed as follows:

I = 1 (first line)

LINEPOINTER(1) = ∅ → print a blank line

I = 2 (second line)

LINEPOINTER(2) = 2 → set column 2, line 2

COLUMNPOINTER(2) = ∅ → no more columns to be set

I = 3 (third line)

LINEPOINTER(3) = 4 → set column 4, line 3

COLUMNPOINTER(4) = 3 → set column 3, line 3

COLUMNPOINTER(3) = ∅ → no more columns to be set

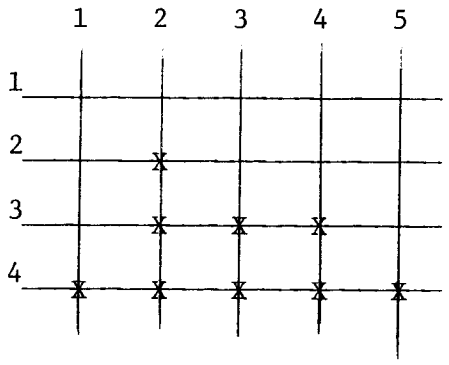
I = 4 (fourth line)

LINEPOINTER(4) = 5 → set column 5, line 4

COLUMNPOINTER(5) = 1 → set column 1, line 4

COLUMNPOINTER(1) = ∅ → no more columns to be set

which would reproduce the following display:



The main loop to create the pointers is:

```
FOR  J:=1  UNTIL NUMBER OF COLUMNS DO
BEGIN
  VALUE:=DATA(J) ;
  COLUMNPOINTER(J):=LINEPOINTER(VALUE) ;
  LINEPOINTER(VALUE):=J ;
END ;
```

Some outputs are shown in Figure 2.