General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of som of the material. However, it is the best reproduction available from the original submission.

DRA

DEPARTMENT OF PHYSICS
SCHOOL OF SCIENCES AND HEALTH PROFESSIONS
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

TECHNICAL REPORT NO. PTR-83-7

DATA REDUCTION PROGRAMS FOR A LASER RADAR SYSTEM

By

F. F. Badavi

and

G. E. Copeland, Principal Investigator

Progress Report
For the period August 1 to October 15, 1983

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

Under
Research Grant NAG-1-1
James M. Hoell, Jr., Technical Monitor

January 1984

DEPARTMENT OF PHYSICS
SCHOOL OF SCIENCES AND HEALTH PROFESSIONS
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

TECHNICAL REPORT NO. PTR-83-7

DATA REDUCTION PROGRAMS FOR A LASER RADAR SYSTEM

By

F. F. Badavi

and

G. E. Copeland, Principal Investigator

Progress Report
For the period August 1 to October 15, 1983

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

Under
Research Grant NAG-1-1
James M. Hoell, Jr., Technical Monitor

Submitted by the
Old Dominion University Research Foundation
P. O. Box 6369
Norfolk, Virginia 23508

January 1984

TABLE OF CONTENTS

LIST OF FIGURES

DATA REDUCTION PROGRAMS
FOR A LASER RADAR SYSTEM

By

F. F. Badavi[1] and G.E. Copeland[2]

SUMMARY

The purpose of this report is to summarize the state of work in the development of software routines, that are used in the pulsed CO_2 laser radar system currently under development at NASA-LARC.

This work was performed under research grant NAG1-1, during the period of August 1 to October 15, 1983.

INTRODUCTION

This report gives the listing and description of software routines which were used to analyze the analog data obtained from LIDAR - system. All routines are written in FORTRAN - IV on a HP - 1000/F mini-computer which serves as the heart of the data acquisition system for the LIDAR program. This particular system has 128 kilobytes of highspeed memory and is equipped with a Vector Instruction Set (VIS) firmware package, which is used in all the routines, to handle quick execution of different long loops. The system handles floating point arithmetic in hardware in order to enhance the speed of execution. This computer is a 2177 C/F series version of HP - 1000 RTE-IVB data acquisition computer system which is designed for real time data capture/analysis and disk/tape mass storage environment.

---

[1]Research Associate, Department of Physics, School of Sciences and Health Professions, Old Dominion University, Norfolk, Virginia.

[2]Associate Professor, Department of Physics, School of Sciences and Health Professions, Old Dominion University, Norfolk, Virginia.

The main programs in this report are LIDAR, TEMPE, and DPLOT. Each program is composed of a main routine which either takes care of initializing all the default values, or does the proper calling sequence to other segmented routines to handle data acquisition, storage, retrieval, computation, smoothing and plotting.

The need for a segmented approach to programming is due to memory limitation of the system in use. The benefit of a segmented program is that it does not have to reside in the main memory until it is needed. Upon calling a swapping between disk and main memory a segmented routine can do its function and be replaced by another segment. This modular approach to programming eliminates the need for large memory requirement.

The remainder of this report will give a brief description of each program, with commented listing of the routine itself at the end. Included are some of the results obtained.

PROGRAM LIDAR

LIDAR, which was partly developed prior to the start of this work, is the main routine for data handling. It accepts digitized data from an A/D converter, (which is the Biomation 1010 digital waveform recorder). This converter provides 10 bit resolution and sends data to the computer in a bit parallel/word serial fashion. During this work. the program and its segments were modified to make them capable of storing data on a 9-track magnetic tape recorder.

The main program LIDAR is written in an interactive fashion, so that the operator can choose the preset default values or enter new values

through the system console. The operator is given the option of plotting the results on a graphics terminal or hardcopy plotter in order to preview the data and decide upon further options such as storage/dump, number of curves/number of frames. Figure 1 is a typical output of this program and its segments for a particular run and it shows the attenuation of returned signal versus amplitude. Four curves were plotted within one frame and the time of the day is written on the upper right corner of graph for each curve. Included is the date and number of averaged shots to obtain the data.

The segmented routines for LIDAR are BISEG, TRASH, POSTP and PTLID. The function of these segments briefly are:

BISEG reads data from Biomation and averages NAVG shots, each consisting of the first 512 points of Biomation memory. The second 512 words are used to determine a baseline level.

TRASH, which is written for the sole purpose of writing a trash file at the end of a good series of data on the magnetic tape, performs all of its functions through a series of EXEC calls. A trash file is usually needed so that the operator would know when the last set of data was taken on the previous run.

POSTP handles all the calls to magnetic tape recorder and generates mechanical motion upon receiving instruction from operator, they include forwarding, backward, skipping of data stored on the tape. This is done by reading the heading of each data file and printing it on the system console, so that the operator can decide where the end of file is.

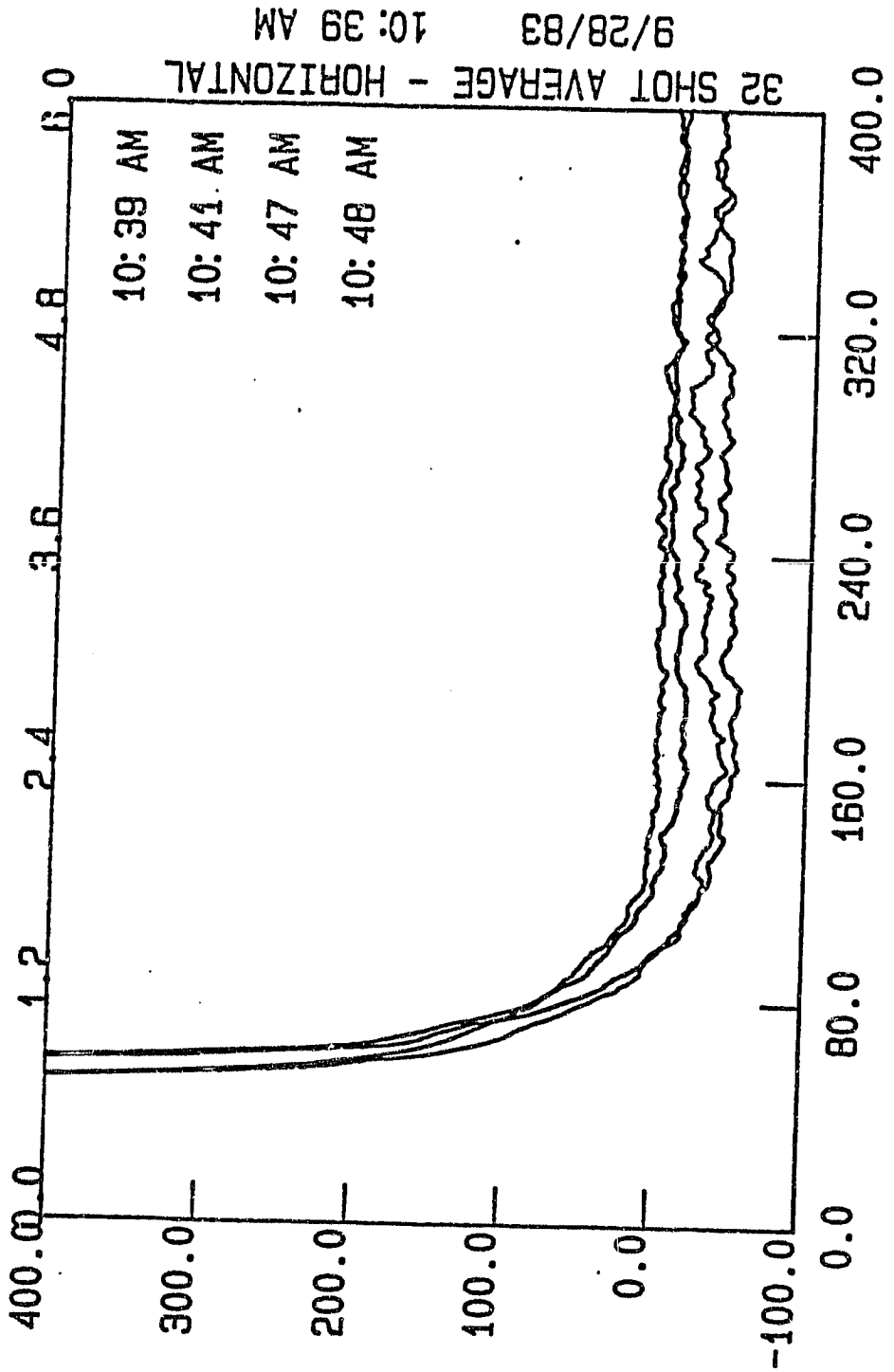

Figure 1. Returned signal amplitude versus distance.

PTLID is the general purpose plotting package for plotting the analyzed data on the plotter/graphic terminal. The plotting is done through many calls to the HP - plot package.

PROGRAM TEMPE

TEMPE is written for the purpose of fast smoothing the stored data on the magnetic tape. The short smoothing technique is applied to all the data points and the result is plotted versus raw data. This program was written when analysis of the stored data was at an early stage and the degree of smoothing was unknown. The program takes advantage of three and five points running averaging techniques to smooth the data. The main pro-program TEMPE has the same modular structure as LIDAR and in many ways is similar to it. It is written in the interactive fashion and allows the operator to select the default values for various plots or input new values.

The segmented routines for TEMPE are FNTEM and PTTEM. The function of these segments briefly are:

FNTEM takes care of operating the tape recorder and positions the tape such that the headings of all the files are printed on the system console. The operator has the option of skipping fow-ward or backward any number of files. The data is then read back to TEMPE and quick smoothing is performed on it.

PTTEM is the general purpose plotting package for plotting the smoothed data versus raw data for the sake of comparison. The routine has the same structure as PTLID in calling HP plot package.

PROGRAM DPLOT

DPLOT is written for the purpose of applying general smoothing techniques to the stored data on the tape. The three smoothing routines are TRIANGULAR, RECTANGULAR and GAUSSIAN. Each one may be applied to the data and the result is plotted along with an analytical fit to the smoothed curve.

The main program DPLOT calls segment REDUC to properly position the tape and gives the operator the option of selecting any one of the three smoothing routines which are included in the subroutine SMOTH. After performing the required task, control is returned back to the main program DPLOT. At this point the program exits to segment PTRED for plotting the smoothed curve and, finally, DAFIT calculates a theoretical fit to the data and plots it through PTRED.

Figures 2(a), 2(b) and 2(c) are typical results obtained. A theoretical fit is plotted along with a smoothed data curve. The vertical lines on the upper x-axis shows the fitting region and information concerning smoothing and fitting is given on the upper right corner.

The segmented routines for DPLOT are REDUC, DAF12 and PTRED. The functions of these segments briefly are:

REDUC handles the operation on the tape recorder and positions the tape such that the headings of all the files are printed on the system console. The operator has the option of skipping forward or backward any number of files. Once this is done a smoothing technique is applied to all the data points.

DAF12 performs a theoretical fit to the data points based

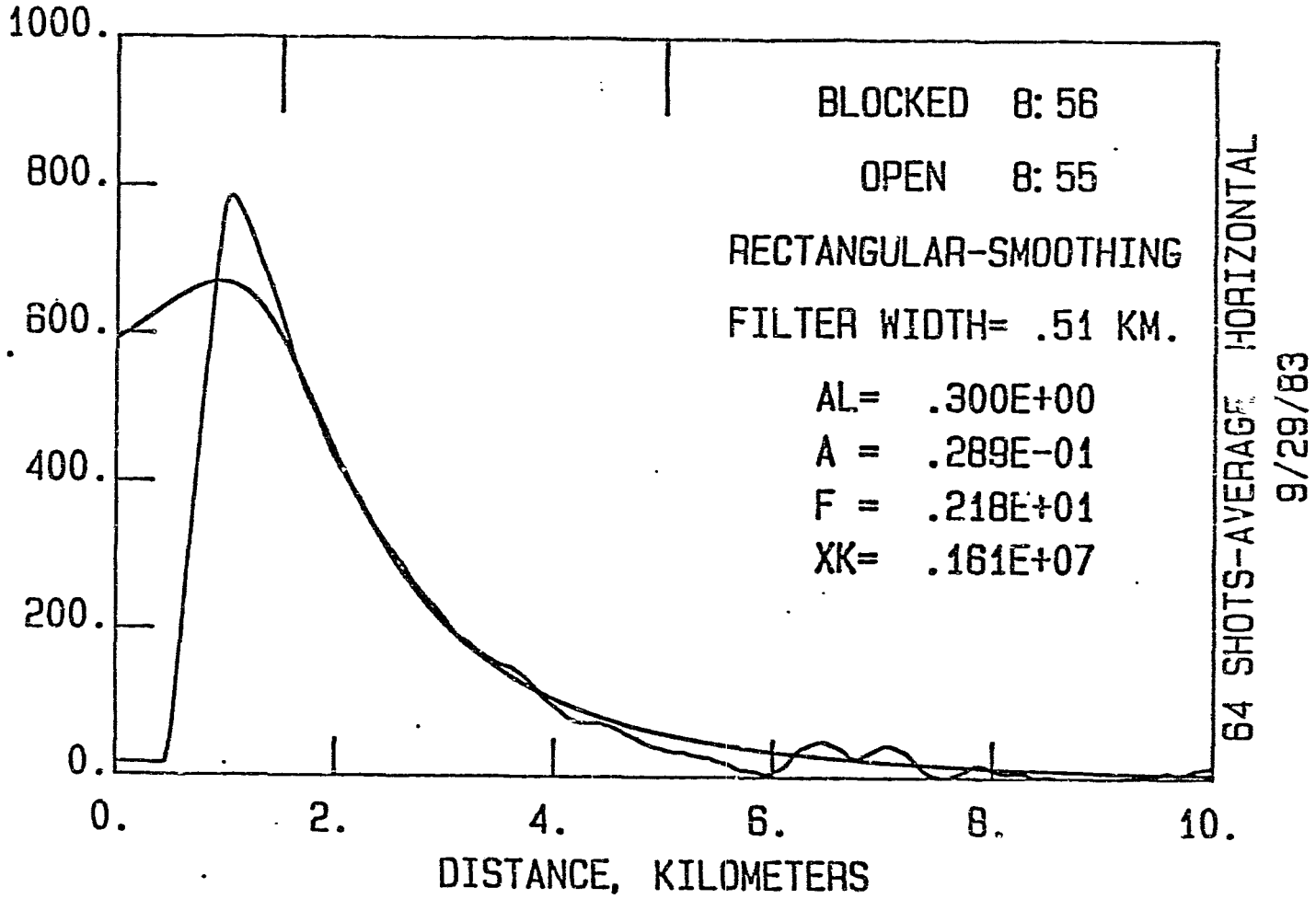

Figure 2(a). A rectangular fit to data.

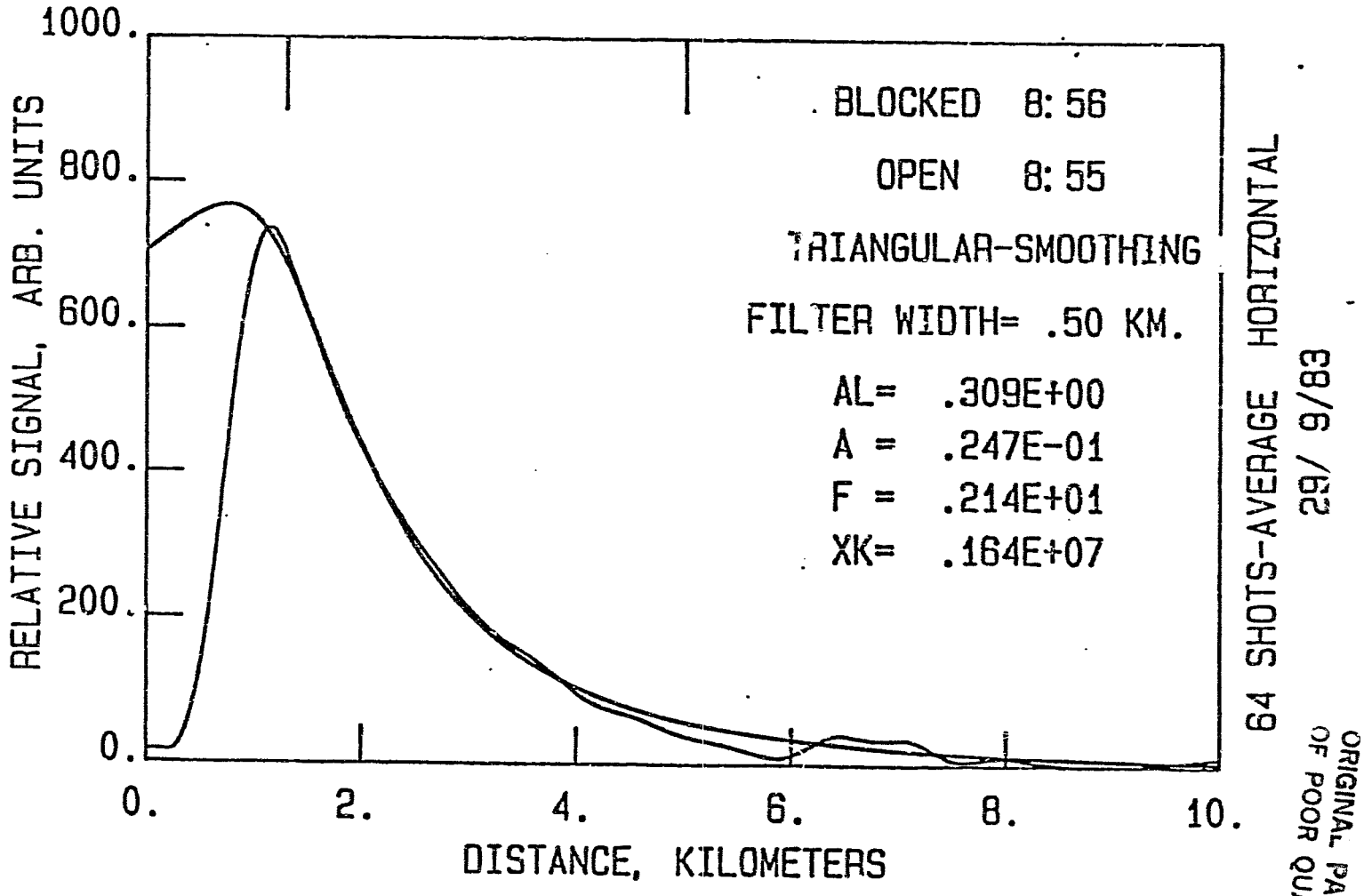

Figure 2(b). A triangular fit to data.

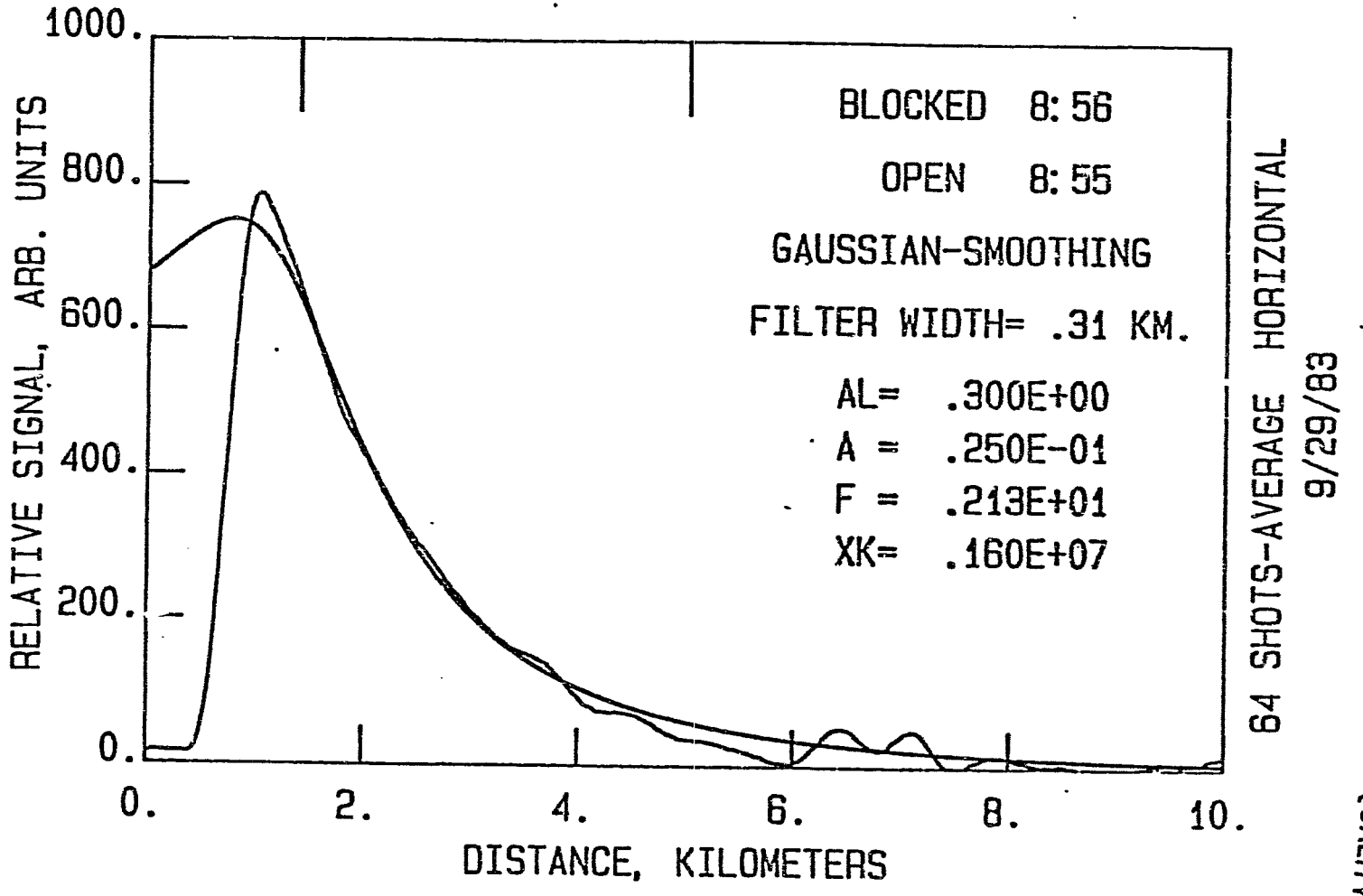

Figure 2(c). A gaussian fit to data.

on the fitting distance which is inputted by the operator.

PTRED is the general purpose plotting package for plotting the smoothed data versus the fit for the sake of comparison. The routine has the same structure as PTLID in calling HP plot package.

ACKNOWLEDGEMENTS

The unlimited and generous comments and suggestions of Clayton H. Bair of NASA - LRC are deeply appreciated.

```
&LIDAR T=00004 IS ON CR00012 USING 00012 BLKS R=0000
0001   FTN,L
0002         PROGRAM LIDAR
0003         COMMON/XNY/X(512),Y(512),ITOUT,NAVG,IFRAM,IOPEN,KFILE
0004         COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0005        -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0006        -,NPLOT,CALIB,XLOOK,IENTRY,MTOPT,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0007         COMMON INSTN1,INSTN2,INSTN3,INSTN4,INSTN5
0008         DIMENSION VPNOM(4,4),IDAYS(12),ITIME(5)
0009         INTEGER BISEG(3),PTLID(3),TRASH(3),POSTP(3)
0010         EQUIVALENCE (IPRAM(1),LU),(IPRAM(2),ID),(IPRAM(3),NAVGDF)
0011         DATA VPNOM/10.,70.,57.,97.,10.,70.,7.,47.,85.,145.,57.,97.,
0012        -85.,145.,7.,47./
0013         DATA BISEG/2HBI,2HSE,2HG /
0014         DATA PTLID/2HPT,2HLI,2HD /
0015         DATA TRASH/2HTR,2HAS,2HH /
0016         DATA POSTP/2HPO,2HST,2HP /
0017   C
0018         DATA IDAYS/31,28,31,30,31,30,31,31,30,31,30,31/
0019   C
0020   C  THIS CALL PICKS UP THE LU AND ID OF THE PLOTTING DEVICE
0021         CALL RMPAR(IPRAM)
0022   C
0023     600 FORMAT(" THIS PRORGAM IS THE MAIN PROGRAM FOR SETTING UP PLOTS",/,
0024        -       " IT READS IN THE PLOT DEVICE LU AND ID THRU RMPAR....",/,
0025        -       " I.E. RU,LIDAR,1,1    PLOTS ON THE CRT  OR",/,
0026        -       "      RU,LIDAR,32,2   PLOTS ON THE PLOTTER",/,
0027        -       "      RU,LIDAR,33,2   PLOTS ON JIM'S PLOTTER",/)
0028     601 FORMAT(" QUESTIONS WILL BE AKSED CONCERNING HOW MANY PLOTS",/,
0029        -       " SCALING OF EACH (INDIVIDUALLY) AND PEN NUMBER",/)
0030         WRITE(1,600)
0031         WRITE(1,601)
0032   C
0033   C  THIS ASSURES THAT A VALID LU/ID COMBINATION IS USED
0034         IF(LU.LE.0 .OR. ID.LE.0) GO TO 90
0035         IF(ID.EQ.1) LU=1
0036         IF(ID.EQ.2.AND.(LU.NE.32.AND.LU.NE.33)) GO TO 90
0037         IF(ID.GT.2) GO TO 90
0038   C
0039     300 FORMAT(1X,"HOW MANY FRAMES DO YOU WANT?",/,"DEFAULT: 4")
0040   C
0041     400 FORMAT(" INPUT VIEWPORT COORD.: VXMIN,VXMAX,VYMIN,VYMAX",/,
0042        -" FOR FRAME",I3,/," DEFAULT VALUES:",3(F3.0,","),F3.0)
0043     401 FORMAT(" INPUT WINDOW COORD. XMIN,XMAX,YMIN,YMAX",/,
0044        -" DEFAULT VALUES: 0,500,-100,900")
0045     402 FORMAT(" HOW MANY CURVES ON FRAME",I2,"?",/,
0046        -"NOTE --- 0 IS AN ABORT",/,
0047        -"          4 IS THE DEFAULT")
0048     403 FORMAT(//10("*"),/"INPUT SAMPLE INTERVAL IN MICROSEC",/,
0049        -" DEFAULT VALUE: .2")
0050     404 FORMAT(//10("*"),/"INPUT 1 FOR HORIZONTAL OR 2 FOR VERTICAL",/,
0051        -" DEFAULT VALUE:  1 (HORIZONTAL)")
0052   C
0053   C
0054     501 FORMAT(" INPUT THE NUMBER OF SHOTS YOU WANT TO AVERAGE",/,
0055        -" NOTE --- 0 IS A  ** CURVE **  ABORT",/,
0056        -"          DEFAULT:",I4)
0057     502 FORMAT(//" **********************************************",/,
0058        -         " *              PLACE PEN IN HOLDER            *",/,
0059        -         " **********************************************",//)
0060     503 FORMAT(" INPUT DEFAULT VALUE FOR NO. OF SHOTS TO BE AVERAGED",/,
0061        -       "   DEFAULT:  32")
0062     504 FORMAT(" DO YOU WANT TO STORE THESE DATA ON MAGNETIC TAPE?",/,
0063        -       "   DEFAULT: N")
0064     505 FORMAT(A1)
```

```
0065     506 FORMAT(//" IS THIS A *** BLOCKED *** CASE?",/,    DEFAULT:  N")
0066     507 FORMAT(" PRESS **  RETURN  ** TO CONTINUE")
0067     508 FORMAT(///2X,30("*"),/,
0068        -       " IF THIS IS THE FIRST RUN OF THE DAY FOR WHICH ",/,
0069        -" YOU ARE PLANNING TO STORE DATA ON TAPE, YOU PROBABLY WANT",/,
0070        -" TO REPOSITION THE TAPE TO THE END OF THE PREVIOUSLY RECORDED"/,
0071        -" DATA.  THIS REQUIRES A LITTLE TIME SO BE PATIENT.",//,
0072        -" DO YOU WANT TO REPOSITION THE TAPE? (Y OR N)",/,)
0073     509 FORMAT(" FOR THIS QUESTION THERE IS NO DEFAULT.  ANSWER Y OR N"/)
0074     510 FORMAT(" POSITIONING TAPE....")
0075         MTCNT=0
0076         IYES=1HY
0077         NO=1HN
0078         ASSIGN 101 TO INSTN1
0079         ASSIGN 102 TO INSTN2
0080         ASSIGN 103 TO INSTN3
0081         ASSIGN 104 TO INSTN4
0082         ASSIGN 105 TO INSTN5
0083         IENTRY=0
0084         NPLOT=0
0085         ITOUT=50
0086         WRITE(1,503)
0087         READ(1,*) NAVGDF
0088         IF(NAVGDF.EQ.0) NAVGDF=32
0089         WRITE(1,300)
0090         READ(1,*) NPLOT
0091         IF(NPLOT.EQ.0) NPLOT=4
0092   C
0093   C
0094         DO 92 NPLT=1,NPLOT
0095         VXMIN=0.
0096         VXMAX=0.
0097         KNOM=MOD(NPLT,4)
0098         IF(KNOM .EQ. 0) KNOM=4
0099         WRITE(1,400) NPLT,(VPNOM(JNOM,KNOM),JNOM=1,4)
0100         READ(1,*) VXMIN,VXMAX,VYMIN,VYMAX
0101         IF(VXMIN.NE.0. .OR. VXMAX.NE.0.) GO TO 30
0102         VXMIN=VPNOM(1,KNOM)
0103         VXMAX=VPNOM(2,KNOM)
0104         VYMIN=VPNOM(3,KNOM)
0105         VYMAX=VPNOM(4,KNOM)
0106      30 CONTINUE
0107   C
0108         XMIN=0.
0109         XMAX=0.
0110         WRITE(1,401)
0111         READ(1,*) XMIN,XMAX,YMIN,YMAX
0112         IF(XMAX.NE.0. .OR. XMIN.NE.0.) GO TO 31
0113         XMIN=0.
0114         XMAX=500.
0115         YMIN=-100.
0116         YMAX=900.
0117      31 CONTINUE
0118   C
0119         XLOOK=0.
0120         WRITE(1,404)
0121         READ(1,*) XLOOK
0122         IF(XLOOK.EQ.0.) XLOOK=1.
0123   C
0124         NCRV=4
0125         WRITE(1,402) NPLT
0126         READ(1,*) NCRV
0127         IF(NCRV.EQ.0) GO TO 32
0128   C
0129   C NCRV = 0 IS AN ABORT
0130   C
```

```
0131          CALIB=0.
0132          WRITE(1,403)
0133          READ(1,*) CALIB
0134          IF(CALIB.EQ.0.) CALIB=.2
0135    C
0136       32 CONTINUE
0137          MCRV=NCRV
0138          IFRAM=1
0139    C
0140          DO 93 ICRV=1,MCRV
0141          IF(NCRV.EQ.0) GO TO 44
0142          NAVG=NAVGDF
0143          WRITE(1,501) NAVGDF
0144          READ(1,*) NAVG
0145          IF(NAVG.EQ.0.AND.ICRV.EQ.MCRV.AND.NPLT.EQ.NPLOT) NCRV=0
0146          IF(NAVG.EQ.0.AND.ICRV.NE.MCRV) GO TO 93
0147          IF(NAVG.EQ.0.AND.ICRV.EQ.MCRV.AND.NPLT.NE.NPLOT) GO TO 93
0148          IF(NCRV.EQ.0) GO TO 44
0149    C
0150    C  NAVG=0 IS A ** CURVE ** ABORT
0151    C
0152          IF(LU.EQ.32) WRITE(1,502)
0153          MTOPT=0
0154          IANS=1HN
0155          WRITE(1,504)
0156          READ(1,505) IANS
0157          IF(IANS.NE.IYES) GO TO 33
0158    C
0159          MTOPT=1
0160          MTCNT=MTCNT+1
0161          WRITE(1,510)
0162          CALL EXEC(8,POSTP)
0163      105 CONTINUE
0164          IANS=1HN
0165          IOPEN=2HOP
0166          WRITE(1,506)
0167          READ(1,505) IANS
0168          IF(IANS.EQ.IYES) IOPEN=2HBL
0169       33 CONTINUE
0170    C
0171    C     IF(IFRAM.NE.1) GO TO 40
0172          CALL EXEC(11,ITIME,IYER)
0173          IAM=2HAM
0174          IF(ITIME(4).GE.12) IAM=2HPM
0175          IF(ITIME(4).GT.12) ITIME(4)=ITIME(4)-12
0176          IHOUR=ITIME(4)
0177          MINUT=ITIME(3)
0178    C
0179          IF(MOD(IYER,4).EQ.0) IDAYS(2)=29
0180          IDUMB=0
0181          DO 35 I=1,12
0182          M7=I
0183          IDUMB=IDAYS(I)+IDUMB
0184          IF(ITIME(5).LE.IDUMB) GO TO 36
0185       35 CONTINUE
0186       36 MON=M7
0187          IDAY=IDAYS(M7)-(IDUMB-ITIME(5))
0188          IYEAR=IYER-1900
0189    C
0190          IF(IFRAM.NE.1) GO TO 40
0191          CALL EXEC(8,PTLID)
0192      103 CONTINUE
0193          IFRAM=0
0194          WRITE(1,507)
0195          READ(1,*) PAUSE
0196       40 CONTINUE
```

```
0197  C
0198        CALL EXEC(8,BISEG)
0199    101 CONTINUE
0200        IF(MTOPT.EQ.1) CALL EXEC(8,TRASH)
0201    104 CONTINUE
0202  C
0203     44 CONTINUE
0204        CALL EXEC(8,PTLID)
0205    102 CONTINUE
0206  C
0207        IF(NCRV.EQ.0) GO TO 41
0208     93 CONTINUE
0209        IF(NCRV.EQ.0) GO TO 41
0210     92 CONTINUE
0211        GO TO 91
0212  C
0213  C
0214     90 CONTINUE
0215        WRITE(1,999)
0216    998 FORMAT(1X,I4," FILES WRITTEN TO TAPE")
0217    999 FORMAT(1X,"IMPROPER LU OR ID...USE PARAMETERS IN RU COMMAND")
0218     91 CONTINUE
0219  C
0220  C
0221     41 CONTINUE
0222        WRITE(1,998) MTCNT
0223        END
0224        BLOCK DATA XYZ
0225        COMMON/XNY/X(512),Y(512),ITOUT,NAVG,IFRAM,IOPEN,KFILE
0226        COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0227       -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0228       -,NPLOT,CALIB,XLOOK,IENTRY,MTOPT,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0229        COMMON INSTN1,INSTN2,INSTN3,INSTN4,INSTN5
0230        END
0231        END$
```

```
&BISEG T=00004 IS ON CR00012 USING 00018 BLKS R=0000

0001  FTN4,L
0002        PROGRAM BISEG(5)
0003        COMMON/XNY/X(512),DATA(512),ITOUT,NAVG,IFRAM,IOPEN,KFILE
0004        COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0005       -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0006       -,NPLOT,CALIB,XLOOK,IENTRY,MTOPT,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0007        COMMON INSTN1,INSTN2,INSTN3,INSTN4,INSTN5
0008        DIMENSION IDATA(1026),IREG(2),ICBUF(5),IBUFF(15)
0009        DIMENSION IWRTB(2),ZERO(512),STORE(512),XDATA(512)
0010        DIMENSION IDAT2(514)
0011        EQUIVALENCE(REG,IREG(1))
0012  C
0013        IBUFF(1)=MON
0014        IBUFF(2)=IDAY
0015        IBUFF(3)=IYEAR
0016        IBUFF(4)=IHOUR
0017        IBUFF(5)=MINUT
0018        IBUFF(6)=IAM
0019        IBUFF(7)=NAVG
0020        IBUFF(8)=XLOOK
0021        IBUFF(9)=10*CALIB
0022        IBUFF(10)=IOPEN
0023        IBUFF(11)=KFILE
0024        IBUFF(12)=0
0025        IBUFF(13)=0
0026        IBUFF(14)=0
0027        IBUFF(15)=0
0028  C
0029  C
0030  C THIS SEGMENT READS DATA FROM THE BIOMATION.  IT AVERAGES
0031  C NAVG SHOTS, EACH CONSISTING OF THE FIRST 512 POINTS OF
0032  C BIOMATION MEMORY.  THE SECOND 512 POINTS OF BIOMATION
0033  C MEMORY ARE USED TO DETERMINE A BASELINE (I.E. ZERO)
0034  C LEVEL.  THE LAST 1024 POINTS ARE DISGARDED TO ENABLE
0035  C READING DATA AT A 2 HERTZ REP RATE.  TRYING TO READ
0036  C ALL 2048 WORDS SLOWS THE TRANSFER TIME TOO MUCH AND
0037  C A 1 HERTZ MAXIMUM IS ABOUT ALL YOU CAN GET FROM THIS
0038  C SEGMENT.  THIS SEGMENT USES THE DVM72 DRIVER SUPPLIED
0039  C BY HP FOR THE 12566B I/F.  REFER TO THE DRIVER MANUAL
0040  C FOR DETAILS CONCERNING THE VARIOUS EXEC CALLS.
0041  C
0042  C THIS SEGMENT IS BASED ON THE STAND-ALONE PROGRAM "YBIOM"
0043  C WHICH IS USED SIMPLY TO READ DATA AND PRINT OUT THE AVERAGE.
0044  C
0045  C
0046  C NCRV=0 IS AN ABORT
0047  C
0048        IF(NCRV.EQ.0) GO TO 41
0049  C
0050        N1026=1026
0051  C
0052  C N IS THE NO. OF DATA WORDS + NO. OF WORDS IN COMMAND STRING
0053  C
0054        ICWTO=133B
0055  C
0056  C  ICWTO IS THE CONTROL WORD FOR SETTING TIME OUT
0057  C  BIT 6 SAYS RESETTING TIME OUT VALUE
0058  C
0059        IF(ITOUT .LE. 20) ITOUT=50
0060  C
0061  C  ITOUT=50 SAYS T.O.VALUE IS TO BE 500 MILLISECONDS
0062  C
0063        CALL EXEC(3,ICWTO,ITOUT)
0064  C
```

```
0065          IDATA(1)=14
0066          ICBL=2                          ORIGINAL PAGE IS
0067          ICBUF(1)=5                      OF POOR QUALITY
0068          ICBUF(2)=6
0069          ICODE=1
0070          ICW2=11533B
0071    C           15  (14 13 12) (11 10 9) (8 7 6) (5 4 3) (2 1 0)
0072    C                      1          1   1   1     1 1     1 1
0073    C                    1          1       5       3       3
0074    C   ICW2 IS THE CONTROL WORD FOR PERFORMING COMMAND SEQ. WITH INTERUPT
0075    C    BIT 12 SAYS USING ICBUF
0076    C    BIT 10 SAYS T. O. IS LEGAL. MUST BE SET FOR IDATA(1)=12
0077    C    BIT 9 SAYS SUBFUNCTION IS IN IDATA(1)
0078    C    BITS 8 AND 6 SAY STANDARD READ.
0079    C
0080          IWRTL=2
0081          IWRTB(1)=1
0082          IWRTB(2)=0
0083          ICWRT=2333B
0084    C       15 (14 13 12) (11 10 9) (8 7 6) (5 4 3) (2 1 0)
0085    C                          1       1 1     1 1     1 1
0086    C                          2       3       3       3
0087    C   ICWRT IS THE CONTROL WORD FOR UNINTERUPTED WRITE
0088    C   BIT 10 SAYS T. O. IS LEGAL
0089    C   BITS 7 AND 6 SAY WRITE WITHOUT INTERUPT UNTIL COMPLETE
0090    C   THE TWO DATA WORDS IN IWRTB EFFECTIVELY ISSUE AN OSR
0091    C
0092    C  ****************************
0093    C
0094    C RESET AFTER EACH DATA TRANSFER USING ICBUF IN READ EXEC CALL
0095    C COMMAND 5 IS A  CLF
0096    C COMMAND 6 IS AN STF
0097    C  SUBFUNCTION 14 ALLOWS CLC, STF IN FINI
0098    C
0099      101 FORMAT(1X,I4,3X,8F6.1,4X,O6,F7.1)
0100          DO 30 I=1,512
0101          STORE(I)=0.0
0102       30 CONTINUE
0103    C *********
0104          L=1
0105    C *********
0106          ICTAP=10B
0107          IF(MTOPT.EQ.1) CALL EXEC(2,ICTAP,IBUFF,15)
0108    C
0109       20 CONTINUE
0110          REG=EXEC(ICODE,ICW2,IDATA,N1026,ICBUF,ICBL)
0111          IAREG=IREG(1)
0112          REG= EXEC(2,ICWRT,IWRTB,IWRTL)
0113    C ********
0114          DO 22 JHU=1,512
0115          XDATA(JHU)=IDATA(JHU+2)
0116       22 CONTINUE
0117    C ********
0118           IDAT2(1)=L
0119           IDAT2(2)=1000
0120          DO 10 I=1,512
0121          DATA(I)=IDATA(I+2)
0122          ZERO(I)=IDATA(I+514)
0123       10 CONTINUE
0124          CALL VSUM(ZBAR,ZERO(1),1,512)
0125          ZBAR=-1.0*ZBAR/512.
0126          CALL VSAD(ZBAR,DATA,1,DATA,1,512)
0127          CALL VADD(DATA,1,STORE,1,STORE,1,512)
0128          WRITE(1,101) L,(DATA(I),I=100,450,50),IAREG,ZBAR
0129    C *********
0130          IF(IAREG.EQ.35004B) GO TO 20
```

```
0131          IF(MTOPT.NE.1) GO TO 32
0132          DO 33 I=3,514
0133           IDAT2(I)=IDATA(I)+ZBAR
0134       33 CONTINUE
0135          CALL EXEC(2,ICTAP,IDAT2,514)
0136       32 CONTINUE
0137          L=L+1
0138          IF(L.LE.NAVG) GO TO 20
0139       21 CONTINUE
0140          ICEOF=110B
0141          IF(MTOPT.EQ.1) CALL EXEC(3,ICEOF)
0142    C ********
0143          XAVG=1.0/FLOAT(NAVG)
0144          CALL VSMY(XAVG,STORE,1,DATA,1,512)
0145    C     DO 40 IJK=1,490,10
0146    C     IMIN=IJK
0147    C     IMAX=IMIN+9
0148    C     WRITE(1,100) (DATA(I),I=IMIN,IMAX)
0149    C  40 CONTINUE
0150       41 CONTINUE
0151      100 FORMAT(10F7.0)
0152          GO TO INSTN1
0153          END
0154          END$
```

```
&TRASH T=00004 IS ON CR00012 USING 00005 BLKS R=0000

0001   FTN4,L
0002         PROGRAM TRASH(5)
0003         COMMON /XNY/X(512),Y(512),ITOUT,NAVG,IFRAM,IOPEN,KFILE
0004         COMMON INSTN1,INSTN2,INSTN3,INSTN4,INSTN5
0005         DIMENSION IBUFF1(15),IBUFF2(514)
0006   C
0007   C  DATE
0008         IBUFF1(1)=13
0009         IBUFF1(2)=39
0010         IBUFF1(3)=83
0011   C
0012   C  TIME
0013         IBUFF1(4)=13
0014         IBUFF1(5)=71
0015         IBUFF1(6)=2HXX
0016   C
0017   C  NUMBER OF SHOTS
0018         IBUFF1(7)=1
0019   C
0020   C HORIZONTAL/VERTICAL
0021         IBUFF1(8)=0
0022   C
0023   C TIME/POINT IN TENTHS OF MICROSECONDS
0024         IBUFF1(9)=0
0025   C
0026   C  OPEN/BLOCKED
0027         IBUFF1(10)=2HXX
0028         IBUFF1(11)=KFILE+1
0029         IBUFF1(12)=0
0030         IBUFF1(13)=0
0031         IBUFF1(14)=0
0032         IBUFF1(15)=0
0033         IBUFF2(1)=1
0034         IBUFF2(2)=999
0035         DO 10 I=3,514
0036         IBUFF2(I)=0
0037      10 CONTINUE
0038         ICNWD=10B
0039         CALL EXEC(2,ICNWD,IBUFF1,15)
0040         CALL EXEC(2,ICNWD,IBUFF2,514)
0041         ICNWD=110B
0042         CALL EXEC(3,ICNWD)
0043         ICBK=1410B
0044         ICFWD=1310B
0045         CALL EXEC(3,ICBK)
0046         CALL EXEC(3,ICBK)
0047         CALL EXEC(3,ICFWD)
0048         GO TO INSTN4
0049         END
0050         END$
```

```
&POSTP T=00004 IS ON CR00012 USING 00005 BLKS R=0000

0001   FTN4,L
0002         PROGRAM POSTP(5)
0003         COMMON /XNY/X(512),Y(512),ITOUT,NAVG,IFRAM,IOPEN,KFILE
0004         COMMON INSTN1,INSTN2,INSTN3,INSTN4,INSTN5
0005         DIMENSION IBUFF1(15),IBUFF2(514)
0006   C
0007         ICNWD=10B
0008         ICFWD=1310B
0009         ICBK=1410B
0010         IREWND=410B
0011   C
0012         WRITE(1,200)
0013     200 FORMAT(26X," SHOTS  H/V  T/P OP/BL")
0014   C     CALL EXEC(3,IREWND)
0015         KOUNT=0
0016      95 CALL EXEC(1,ICNWD,IBUFF1,15)
0017         KFILE=IBUFF1(11)
0018         KOUNT=KOUNT+1
0019         WRITE(1,101) KOUNT,(IBUFF1(I),I=1,15)
0020     101 FORMAT(2X,2I4,"/",I2,"/",I2,2X,I2,":",I2,1X,A2,3I5,3X,A2,5I5)
0021   C
0022   C    CHECK TO SEE IF TRASH FILE. I.E., MONTH AND HOUR ARE 13.
0023   C
0024         IF(IBUFF1(1).EQ.13 .AND. IBUFF1(4).EQ.13) GO TO 104
0025         C.  L EXEC(3,ICFWD)
0026         GO TO 95
0027     104 CONTINUE
0028         CALL EXEC(3,ICBK)
0029         CALL EXEC(3,ICFWD)
0030         GO TO INSTN5
0031         END
0032         END$
```

```
&PTLID T=00004 IS ON CR00012 USING 00005 BLKS R=0000

0001   FTN,L
0002         PROGRAM PTLID(5)
0003         COMMON/XNY/X(512),Y(512),ITOUT,NAVG,IFRAM,IOPEN,KFILE
0004         COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0005        -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0006        -,NPLOT,CALIB,XLOOK,IENTRY,MTOPT,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0007         COMMON INSTN1,INSTN2,INSTN3,INSTN4,INSTN5
0008         DIMENSION DIST(6)
0009     100 FORMAT("SP0")
0010     101 FORMAT("SP1")
0011     102 FORMAT("SP2")
0012     103 FORMAT("SP3")
0013     104 FORMAT("SP4")
0014     105 FORMAT("SP5")
0015     106 FORMAT("SP6")
0016     107 FORMAT("SP7")
0017     108 FORMAT("SP8")
0018         IENTRY=IENTRY+1
0019         NP=512
0020         DO 10 I=1,NP
0021         X(I)=I
0022      10 CONTINUE
0023         LU=IPRAM(1)
0024         ID=IPRAM(2)
0025         IF(ID.EQ.1.AND.IENTRY.EQ.1)
0026        -CALL PLOTR(IGCB,ID,4,LU,IOBUF,20)
0027         IF(ID.EQ.2.AND.IENTRY.EQ.1)
0028        -CALL PLOTR(IGCB,ID,4,LU,IOBUF,20)
0029   C
0030   C  NCRV.EQ.0 IS AN ABORT
0031   C
0032         IF(NCRV.EQ.0) GO TO 41
0033   C
0034         IP=MOD(ICRV,8)
0035         IF(IP.LE.0) IP=8
0036   C     IF(IP .EQ. 1.AND.LU.NE.32) WRITE(LU,101)
0037         IF(IP .EQ. 1) WRITE(LU,101)
0038         IF(IP .EQ. 2) WRITE(LU,102)
0039         IF(IP .EQ. 3) WRITE(LU,103)
0040         IF(IP .EQ. 4) WRITE(LU,104)
0041         IF(IP .EQ. 5) WRITE(LU,105)
0042         IF(IP .EQ. 6) WRITE(LU,106)
0043         IF(IP .EQ. 7) WRITE(LU,107)
0044         IF(IP .EQ. 8) WRITE(LU,108)
0045   C     IF(IP .EQ. 2.AND.LU.NE.32) WRITE(LU,102)
0046   C     IF(IP .EQ. 3.AND.LU.NE.32) WRITE(LU,103)
0047   C     IF(IP .EQ. 4.AND.LU.NE.32) WRITE(LU,104)
0048   C     IF(IP .EQ. 5.AND.LU.NE.32) WRITE(LU,105)
0049   C     IF(IP .EQ. 6.AND.LU.NE.32) WRITE(LU,106)
0050   C     IF(IP .EQ. 7.AND.LU.NE.32) WRITE(LU,107)
0051   C     IF(IP .EQ. 8.AND.LU.NE.32) WRITE(LU,108)
0052         CALL SETAR(IGCB,1.5)
0053         CALL VIEWP(IGCB,VXMIN,VXMAX,VYMIN,VYMAX)
0054         CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
0055         HGT=3.0
0056         AR=0.4
0057         SANG=0.0
0058         ICOOR=0
0059         CALL CSIZE(IGCB,HGT,AR,SANG,ICOOR)
0060         CALL FXD(IGCB,1)
0061         XTIC=(XMIN-XMAX)/5.
0062         YTIC=(YMAX-YMIN)/5.
0063         IF(IFRAM.NE.1) GO TO 42
0064         CALL FRAME(IGCB)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0065          CALL FRAME(IGCB)
0066          CALL LAXES(IGCB,XTIC,YTIC,XMIN,YMIN,1.,1.,2.)
0067          VXMI3=VXMIN-3.
0068          VXMA3=VXMAX+6.
0069          VYMI3=VYMIN-3.
0070          VYMA3=VYMAX+3.
0071          CALL VIEWP(IGCB,VXMI3,VXMA3,VYMI3,VYMA3)
0072          CALL WINDW(IGCB,VXMI3,VXMA3,VYMI3,VYMA3)
0073          DO 43 IDIST=1,6
0074          DIST(IDIST)=(XMIN-(IDIST-1)*XTIC)*CALIB*.3/2
0075       43 CONTINUE
0076          CALL LORG(IGCB,1)
0077          CALL MOVE(IGCB,VXMI3,VYMAX)
0078          CALL LABEL(IGCB)
0079          WRITE(LU,431) (DIST(IDIST),IDIST=1,6)
0080      431 FORMAT(1X,5(F4.1,6X),F4.1)
0081    C
0082          VXMA2=VXMA3-3.
0083          CALL MOVE(IGCB,VXMA2,VYMIN)
0084          CALL LDIR(IGCB,1.57)
0085          IF(XLOOK.EQ.2) GO TO 30
0086          CALL LABEL(IGCB)
0087          WRITE(LU,432) NAVG,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0088          GO TO 31
0089       30 CONTINUE
0090          CALL LABEL(IGCB)
0091          WRITE(LU,433) NAVG,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0092       31 CONTINUE
0093          CALL LDIR(IGCB,0.)
0094      432 FORMAT(I4," SHOT AVERAGE - HORIZONTAL",/,
0095         -6X,I2,"/",I2,"/",I2,4X,I2,":",I2,1X,A2)
0096      433 FORMAT(I4," SHOT AVERAGE -  VERTICAL ",/,
0097         -6X,I2,"/",I2,"/",I2,4X,I2,":",I2,1X,A2)
0098          CALL MOVE(IGCB,VXMIN,VYMIN)
0099          CALL VIEWP(IGCB,VXMIN,VXMAX,VYMIN,VYMAX)
0100          CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
0101    C
0102    C
0103          IF(IFRAM.EQ.1) GO TO INSTN3
0104       42 CONTINUE
0105          DO 20 J=1,NP
0106          IF(J.GT.1) GO TO 21
0107          CALL MOVE(IGCB,X(J),Y(J))
0108          GO TO 20
0109       21 CALL DRAW(IGCB,X(J),Y(J))
0110       20 CONTINUE
0111          XTIME=XMIN+3.2*(XMAX-XMIN)/5.
0112          YTIME=YMAX-ICRV*(YMAX-YMIN)/10.
0113          CALL MOVE(IGCB,XTIME,YTIME)
0114          CALL LABEL(IGCB)
0115          IF(MTOPT.NE.1) WRITE(LU,201) IHOUR,MINUT,IAM
0116          IF(MTOPT.EQ.1) WRITE(LU,202) IHOUR,MINUT,IAM,KFILE
0117      201 FORMAT(1X,I2,":",I2,1X,A2)
0118      202 FORMAT(1X,I2,":",I2,1X,A2,1X,I4)
0119          CALL XMIT(IGCB)
0120       41 CONTINUE
0121          IF(NPLT.EQ.NPLOT.AND.ICRV.EQ.NCRV) WRITE(LU,100)
0122          IF(NCRV.EQ. 0) WRITE(LU,100)
0123          IF(NPLT.EQ.NPLOT.AND.ICRV.EQ.NCRV) CALL PLOTR(IGCB,ID,0)
0124          IF(NCRV.EQ. 0) CALL PLOTR(IGCB,ID,0)
0125          GO TO INSTN2
0126          END
0127          END$
```

```
&TEMPE T=00004 IS ON CR AA   USING 00030 BLKS R=0000

0001   FTN4,L
0002         PROGRAM TEMPE
0003         COMMON/XNY/NAVG,IFRAM,IBUF1(15),IBUF2(514)
0004         COMMON IYES,NO
0005         COMMON/ITICS/ITIC1,ITIC2,ITIC3,ITIC4,ITIC5,ITIC6
0006         COMMON IREWND,ICNWD,ICNWD1,ICNWD2,ICNWD3,NPTS
0007         COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3),
0008        -XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT,
0009        -NPLOT,CALIB,IENTRY,IAM
0010         COMMON INSTN2,INSTN3,INSTN4,INSTN5
0011         DIMENSION VPNOM(4,4),IDAYS(12),ITIME(5)
0012         INTEGER PTTEM(3),FNTEM(3)
0013         EQUIVALENCE(IPRAM(1),LU),(IPRAM(2),ID)
0014         DATA VPNOM/10.,70.,57.,97.,10.,70.,7.,47.,85.,145.,57.,97.,
0015        -85.,145.,7.,47./
0016         DATA IDAYS/31,28,31,30,31,30,31,31,30,31,30,31/
0017         IREWND=410B
0018         ICNWD=10B
0019         ICNWD1=1310B
0020         ICNWD2=1410B
0021         ICNWD3=110B
0022         NPTS=514
0023         ITIC1=1H:
0024         ITIC2=1H-
0025         ITIC3=1HH
0026         ITIC4=1HV
0027         ITIC5=2HOP
0028         ITIC6=2HEL
0029         IYES=1HY
0030         NO=1HN
0031         ASSIGN 102 TO INSTN2
0032         ASSIGN 103 TO INSTN3
0033         ASSIGN 104 TO INSTN4
0034         ASSIGN 105 TO INSTN5
0035   C     THIS CALL PICKS UP THE LU AND ID OF THE PLOTTING DEVICE
0036         CALL RMPAR(IPRAM)
0037   C     THIS ASSURES THAT A PROPER LU/ID COMBINATION IS USED
0038         IF(LU.LE.0.OR.ID.LE.0)GO TO 90
0039         IF(ID.EQ.1)LU=1
0040         IF(ID.EQ.2.AND.LU.NE.32)GO TO 90
0041         IF(ID.GT.2)GO TO 90
0042         WRITE(1,600)
0043      68 FORMAT(A1)
0044      42 FORMAT(" YOU MUST ANSWER Y(YES) OR N(NO) ONLY !!, TRY AGAIN")
0045     600 FORMAT(" THIS PROG.IS THE MAIN PROG.FOR SETTING UP PLOTS",/,
0046        -       " IT READS IN THE PLOT DEVICE LU AND ID THRU. RMPAR....",/,
0047        -       " I. E.  RU,STATS,1,1   PLOTS ON THE CRT OR",/,
0048        -       "        RU,STATS,32,2   PLOTS ON THE PLOTTER",/)
0049         WRITE(1,601)
0050     601 FORMAT(" QUESTIONS WILL BE ASKED CONCERNING HOW MANY PLOTS,",/,
0051        -       " SCALING OF EACH INDIVIDUAL PLOT AND PEN NUMBER")
0052         WRITE(1,602)
0053     602 FORMAT(" THIS MAIN PROG.ASSUMES THAT THE ACTUAL PLOTTING",/,
0054        -       " ROUTINE AND TAPE POSITIONING ARE PROGRAM SEGMENTS",/)
0055         WRITE(1,603)
0056     603 FORMAT(" THE MAIN PROG.STATS WILL FIRST REWIND THE MAG. TAPE",/,
0057        -       " AND THEN PRINT OUT HEADINGS OF ALL THE DATA FILES ",/,
0058        -       " ON TAPE, UP TO A TRASH FILE.",/,
0059        -       " YOU MUST INPUT THE FILE# OF THE FILE YOU WISH TO WORK",/,
0060        -       " ON, THRU. THE KEYBOARD TO ACCESS THAT FILE.")
0061     351 CONTINUE
0062         FNTEM(1)=2HFN
0063         FNTEM(2)=2HTE
0064         FNTEM(3)=2HM
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0065          CALL EXEC(8,FNTEM)
0066      104 CONTINUE
0067          CALL EXEC(1,ICNWD,IBUF1,15)
0068          NAVG=IBUF1(7)
0069          CALIB=FLOAT(IBUF1(9))/10.
0070          WRITE(1,273)
0071      273 FORMAT(/," THE FOLLOWING IS HEADING OF YOUR DATA FILE",/)
0072          WRITE(1,274)
0073      274 FORMAT(8X,"#SHOT",4X,"DATE",6X,"TIME",6X,"H/V",7X,
0074         -" T/P",7X,"OP/BL",/,66("="),/)
0075          IF(IBUF1(8).EQ.1)IBUF1(8)=ITIC3
0076          IF(IBUF1(8).EQ.2)IBUF1(8)=ITIC4
0077          IF(IBUF1(10).EQ.1)IBUF1(10)=ITIC5
0078          IF(IBUF1(10).EQ.2)IBUF1(10)=ITIC6
0079          WRITE(1,275)IBUF1(7),IBUF1(1),ITIC2,IBUF1(2),ITIC2,IBUF1(3),
0080         -IBUF1(4),ITIC1,IBUF1(5),IBUF1(6),IBUF1(8),IBUF1(9),IBUF1(10)
0081      275 FORMAT(10X,I4,3X,I2,A1,I2,A1,I2,3X,I2,A1,I2,1X,A2,
0082         -4X,A1,8X,I3,9X,A2)
0083  C
0084  C
0085          WRITE(1,507)
0086          READ(1,*)PAUSE
0087  C
0088  C
0089          DO 20 J=1,NAVG
0090          CALL EXEC(1,ICNWD,IBUF2,514)
0091          DO 60 I=1,514
0092          BUF2(I)=FLOAT(IBUF2(I))
0093       60 CONTINUE
0094          CALL VADD(BUFSU,1,BUF2,1,BUFSU,1,514)
0095       20 CONTINUE
0096          CALL EXEC(3,ICNWD1)
0097          CALL VSMY(1./NAVG,BUFSU,1,BUFSU,1,514)
0098          IENTRY=0
0099          NPLOT=0
0100          WRITE(1,300)
0101      300 FORMAT(" HOW MANY FRAMES/SCREEN DO YOU WISH TO HAVE? <DEFAULT=4>")
0102          READ(1,*)NPLOT
0103          IF(NPLOT.EQ.0)NPLOT=4
0104          PTTEM(1)=2HPT
0105          PTTEM(2)=2HTE
0106          PTTEM(3)=2HM
0107          DO 92 NPLT=1,NPLOT
0108          IF(NPLT.EQ.1)GO TO 401
0109      820 WRITE(1,106)
0110      106 FORMAT(" WISH TO ANALYZE ANOTHER PORTION OF SAME FILE (Y/N)?")
0111          READ(1,68)IANS
0112          IF(IANS.EQ.IYES)GO TO 401
0113          IF(IANS.EQ.NO)GO TO 318
0114          IF((IANS.NE.IYES).OR.(IANS.NE.NO))WRITE(1,42)
0115          GO TO 820
0116      401 VXMIN=0.
0117          VXMAX=0.
0118          KNOM=MOD(NPLT,4)
0119          IF(KNOM.EQ.0)KNOM=4
0120          WRITE(1,400)NPLT,(VPNOM(JNOM,KNOM),JNOM=1,4)
0121      400 FORMAT(" INPUT VIEWPORT COORD.:VXMIN,VXMAX,VYMIN,VYMAX",/,
0122         -".FOR FRAME ",I3," <DEFAULT VALUES=",3(F3.0,","),F3.0,">")
0123          READ(1,*)VXMIN,VXMAX,VYMIN,VYMAX
0124          IF(VXMIN.NE.0..OR.VXMAX.NE.0.)GO TO 30
0125          VXMIN=VPNOM(1,KNOM)
0126          VXMAX=VPNOM(2,KNOM)
0127          VYMIN=VPNOM(3,KNOM)
0128          VYMAX=VPNOM(4,KNOM)
0129       30 CONTINUE
0130          XMIN=0.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0131          XMAX=500.
0132          YMIN=-100.
0133          YMAX=900.
0134       31 CONTINUE
0135          ALTMX=(XMIN-5*((XMIN-XMAX)/5.))*CALIB*.3/2
0136       38 WRITE(1,41)ALTMX
0137       41 FORMAT(/," THE MAX. ALT. FOR THIS FILE IS=",F5.2," KM.",/)
0138          WRITE(1,22)
0139       22 FORMAT(" WHAT RANGE OF ALT. YOU WISH TO LOOK AT <IN KM.> I.E",
0140         -" (1.5,3.5)")
0141          READ(1,*)ALTM1,ALTM2
0142          IF((ALTM1.GE.0.).OR.(ALTM2.LE.ALTMX))GO TO 45
0143          WRITE(1,44)
0144       44 FORMAT(/," ALT. RANGE ERROR !!, TRY AGAIN")
0145          GO TO 38
0146       45 NALTM1=IFIX(FLOAT(NPTS-2)*(ALTM1/ALTMX))
0147          NALTM2=IFIX(FLOAT(NPTS-2)*(ALTM2/ALTMX))
0148          XMIN=(FLOAT(NALTM1)/FLOAT(NPTS-2))*500.
0149          XMAX=(FLOAT(NALTM2)/FLOAT(NPTS-2))*500.
0150          WRITE(1,107)NALTM1,NALTM2,XMIN,XMAX,CALIB
0151      107 FORMAT(2X,I5,5X,I5,5X,F6.0,5X,F6.0,5X,F6.3)
0152          WRITE(1,507)
0153          READ(1,*)PAUSE
0154          NCRV=2
0155          MCRV=NCRV
0156          DO 93 ICRV=1,MCRV
0157          WRITE(1,776) ICRV,MCRV
0158      776 FORMAT(" ICRV=",I5,"  MCRV=",I5)
0159          IF(ICRV.EQ.2)CALL SMOOT(BUFSU,NPTS,ICRV)
0160          IF(LU.EQ.32)WRITE(1,507)
0161      507 FORMAT(" ** PRESS RETURN TO CONTINUE **")
0162          READ(1,*)PAUSE
0163          IF(ICRV.NE.1)GO TO 77
0164          CALL EXEC(11,ITIME,IYER)
0165          IAM=2HAM
0166          IF(ITIME(4).GE.12)IAM=2HPM
0167          IF(ITIME(4).GT.12)ITIME(4)=ITIME(4)-12
0168          IHOUR=ITIME(4)
0169          MINUT=ITIME(3)
0170          IF(MOD(IYER,4).EQ.0)IDAYS(2)=29
0171          IDUMB=0
0172          DO 397 I=1,12
0173          M7=I
0174          IDUMB=IDAYS(I)+IDUMB
0175          IF(ITIME(5).LE.IDUMB)GO TO 378
0176      397 CONTINUE
0177      378 MON=M7
0178          IDAY=IDAYS(M7)-(IDUMB-ITIME(5))
0179          IYEAR=IYER-1900
0180          IF(ICRV.NE.1)GO TO 77
0181          IFRAM=1
0182          CALL EXEC(8,PTTEM)
0183      103 CONTINUE
0184          IFRAM=0
0185       77 IF(ICRV.EQ.1)GO TO 93
0186          CALL EXEC(8,PTTEM)
0187      102 CONTINUE
0188       93 CONTINUE
0189       92 CONTINUE
0190      318 WRITE(1,319)
0191      319 FORMAT(/," DO YOU WISH TO ANALYZE ANOTHER SET (Y/N)?")
0192          READ(1,68)IANS
0193          IF(IANS.EQ.IYES)GO TO 351
0194          IF(IANS.EQ.NO)GO TO 91
0195          IF((IANS.NE.IYES).AND.(IANS.NE.NO))WRITE(1,42)
0196          GO TO 318
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0197      90 CONTINUE
0198         WRITE(1,999)
0199     999 FORMAT(" IMPROPER LU OR ID.... USE PARAMETERS IN RU COMMAND")
0200      91 CONTINUE
0201         END
0202         BLOCK DATA XYZ
0203         COMMON/XNY/NAVG,IFRAM,IBUF1(15),IBUF2(514),
0204        -BUFSU(514),KOUNT,IREWND,ICNWD,ICNWD1,ICNWD2,ICNWD3,NPTS,X(514)
0205         COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3),
0206        -XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT,
0207        -NPLOT,CALIB,IENTRY,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0208         COMMO' TNSTN2,INSTN3,INSTN4,INSTN5
0209         END
0210         SUBROUTINE SMOOT(Y1,N,IP)
0211         DIMENSION Y1(514)
0212         IMAX=N-1
0213         YI=Y1(1)
0214         DO 5 I=2,IMAX
0215   C     IF(IP.EQ.2)YNEW=(YI+2.*Y1(I)+Y1(I+1))/4.
0216         IF(IP.EQ.2)YNEW=((11./32.)*YI+(11./32.)*Y1(I)+(1./4.)*Y1(I+1)
0217        -+(1./16.)*Y1(I+2))
0218         YI=Y1(I)
0219         Y1(I)=YNEW
0220       5 CONTINUE
0221   C     IF(IP.EQ.2)Y1(N)=(YI+3.*Y1(N))/4.
0222         IF(IP.EQ.2)Y1(N)=((1./16.)*Y1(I-2)+(1./4.)*Y1(I-1)+(11./32.)*Y1(I)
0223        -+(11./32.)*Y1(N))
0224         RETURN
0225         END
0226         END$
```

```
&FNTEM T=00004 IS ON CR AA   USING 00012 BLKS R=0000
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0001   FTN4,L
0002         PROGRAM FNTEM(5)
0003         COMMON IYES,NO
0004         COMMON/ITICS/ITIC1,ITIC2,ITIC3,ITIC4,ITIC5,ITIC6
0005         COMMON/XNY/IBUF1(15),X(512),Y(5129,SUM(512),NAVG
0006         COMMON IREWND,ICNWD,ICNWD1,ICNWD2,ICNWD3
0007         COMMON INSTN2,INSTN3,INSTN4,INSTN5
0008         COMMON XJUNK(10)
0009         DIMENSION IDAY(2),IMON(2),IYEAR(2),IHOUR(2),IMIN(2)
0010   C
0011     100 FORMAT(//,8X,"#SHOTS",4X,"DATE",6X,"TIME",6X,"H/V",7X,
0012        -" T/P",7X,"OP/BL",/,66("="),/)
0013     101 FORMAT(10X,I4,3X,I2,A1,I2,A1,I2,3X,I2,A1,I2,1X,A2,4X,A1,8X,I3,
0014        -9X,A2)
0015     103 FORMAT("  SKIPPING ||||.....")
0016   C
0017     200 FORMAT(/," HOW MANY FILES SHOULD I SKIP?")
0018     202 FORMAT(" SHOULD I REWIND THE TAPE?  (Y/N)?")
0019      68 FORMAT(A1)
0020     204 FORMAT(/," HOW MANY FILES SHOULD I BACKSPACE?")
0021      42 FORMAT(" YOU MUST ANSWER Y(YES) OR N(NO) ONLY !!, TRY AGAIN")
0022     900 FORMAT(" FILE READ TERMINATED *** TRASH FILE ENCOUNTERED")
0023   C
0024      14 WRITE(1,202)
0025         READ(1,68) IANS
0026         IF(IANS.EQ.IYES)GO TO 351
0027         IF(IANS.EQ.NO)GO TO 91
0028         IF((IANS.NE.IYES).AND.(IANS.NE.NO))WRITE(1,42)
0029         GO TO 14
0030     351 CALL EXEC(3,IREWND)
0031      91 WRITE(1,200)
0032         READ(1,*) NSKIP
0033         IF(NSKIP.EQ.0) GO TO 11
0034         WRITE(1,103)
0035         WRITE(1,100)
0036         NFILE=0
0037         DO 10 I=1,NSKIP
0038         CALL EXEC(1,ICNWD,IBUF1,15)
0039      96 IF(IBUF1(8).EQ.1)IBUF1(8)=ITIC3
0040         IF(IBUF1(8).EQ.2)IBUF1(8)=ITIC4
0041         IF(IBUF1(10).EQ.1)IBUF1(10)=ITIC5
0042         IF(IBUF1(10).EQ.2)IBUF1(10)=ITIC6
0043         WRITE(1,101)IBUF1(7),IBUF1(1),ITIC2,IBUF1(2),ITIC2,IBUF1(3),
0044        -IBUF1(4),ITIC1,IBUF1(5),IBUF1(6),IBUF1(8),IBUF1(9),IBUF1(10)
0045         IF(NFILE.NE.0)GO TO 97
0046         IF(IBUF1(1).EQ.13 ) GO TO 99
0047         CALL EXEC(3,ICNWD1)
0048      10 CONTINUE
0049      11 CONTINUE
0050   C
0051         WRITE(1,204)
0052         READ(1,*) NBACK
0053         IF(NBACK.EQ.0) GO TO 13
0054         NBACK1=NBACK+1
0055         DO 12 I=1,NBACK1
0056         CALL EXEC(3,ICNWD2)
0057      12 CONTINUE
0058         CALL EXEC(3,ICNWD1)
0059      13 CONTINUE
0060         DO 14 I=1,512
0061         X(I)=0.0
0062      14 CONTINUE
0063         WRITE(1,207)
0064     207 FORMAT(" INPUT THE TIME FOR THE BLOCKED CASE IN THE FORM",/,
```

```
0065        -       " XX<HR.>,YY<MIN.>.......I.E. 10,15")
0066         READ(1,*)IHOUR(1),IMIN(1)
0067         XJUNK(1)=IHOUR(1)
0068         XJUNK(2)=IMIN(1)
0069         WRITE(1,208)
0070
0071
0072
0073
0074     208 FORMAT(" INPUT THE TIME FOR THE OPEN CASE IN THE FORM",/,
0075        -       " XX<HR.>,YY<MIN.>.......I.E. 10,47")
0076         XJUNK(3)=IHOUR(2)
0077         XJUNK(4)=IMIN(2)
0078         NFILE=2
0079         WRITE(1,100)
0080         DO 30 K=1,NFILE
0081         CALL EXEC(1,ICNWD,IBUF1,15)
0082         GO TO 96
0083      97 IF(IBUF1(1).EQ.13.AND.IBUF1(4).EQ.13)GO TO 99
0084         IF(IBUF1(4).EQ.IHOUR(2).AND.IBUF1(5).EQ.IMIN(2))GO TO 16
0085         IF(IBUF1(4).EQ.IHOUR(1).AND.IBUF1(5).EQ.IMIN(1))GO TO 16
0086         GO TO 24
0087      16 CONTINUE
0088         NAVG=IBUF1(7)
0089         DO 15 I=1,512
0090         SUM(I)=0.0
0091      15 CONTINUE
0092         DO 20 J=1,NAVG
0093         CALL EXEC(1,ICNWD,IBUF2,514)
0094         DO 18 I=3,514
0095         SUM(I-2)=SUM(I-2)+IBUF2(I)
0096      18 CONTINUE
0097      20 CONTINUE
0098         IF(IBUF1(4).NE.IHOUR(2).OR.IBUF1(5).NE.IMIN(2))GO TO 22
0099         IDAY(2)=IBUF1(1)
0100         IMON(2)=IBUF1(2)
0101         IYEAR(2)=IBUF1(3)
0102         CALL VSMY(1./NAVG,SUM,1,Y,1,512)
0103      22 CONTINUE
0104         IF(IBUF1(4).NE.IHOUR(1).OR.IBUF1(5).NE.IMIN(1))GO TO 14
0105         IDAY(1)=IBUF1(1)
0106         IMON(1)=IBUF1(2)
0107         IYEAR(1)=IBUF1(3)
0108         CALL VSMY(1./NAVG,SUM,1,X,1,512)
0109      24 CONTINUE
0110         CALL EXEC(3,ICNWD1)
0111      30 CONTINUE
0112         WRITE(1,209)
0113   C 209
0114   C
0115   C
0116   C
0117   C
0118   C
0119   C
0120   C
0121   C
0122   C
0123   C
0124   C
0125   C
0126   C
0127         GO TO 999
0128      99 CONTINUE
0129         WRITE(1,900)
0130         CALL EXEC(3,ICNWD2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0131          CALL EXEC(3,ICNWD1)
0132          STOP
0133      999 CONTINUE
0134          GO TO INSTN4
0135          END
0136          END$
```

```
&PTTEM T=00004 IS ON CR AA   USING 00018 BLKS R=0000

0001  FTN,L
0002        PROGRAM PTTEM(5)
0003        COMMON/XNY/NAVG,IFRAM,IBUF1(15),
0004       -BUFSU(514),KOUNT,IREWND,ICNWD,ICNWD1,ICNWD2,ICNWD3,NPTS,X(514)
0005        COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0006       -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0007       -,NPLOT,CALIB,IENTRY,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0008        COMMON INSTN2,INSTN3,INSTN4,INSTN5
0009        DIMENSION DIST(6)
0010    100 FORMAT("SP0")
0011    101 FORMAT("SP1")
0012    102 FORMAT("SP2")
0013    103 FORMAT("SP3")
0014    104 FORMAT("SP4")
0015    105 FORMAT("SP5")
0016    106 FORMAT("SP6")
0017    107 FORMAT("SP7")
0018    108 FORMAT("SP8")
0019        IENTRY=IENTRY+1
0020        DO 10 I=1,NPTS
0021        X(I)=I
0022     10 CONTINUE
0023        LU=IPRAM(1)
0024        ID=IPRAM(2)
0025        IF(ID.EQ.1.AND.IENTRY.EQ.1)
0026       -CALL PLOTR(IGCB,ID,4,LU,IOBUF,20)
0027        IF(ID.EQ.2.AND.IENTRY.EQ.1)
0028       -CALL PLOTR(IGCB,ID,4,LU,IOBUF,20)
0029        IP=MOD(ICRV,8)
0030        IF(IP.LE.0) IP=8
0031        IF(IP .EQ. 1) WRITE(LU,101)
0032        IF(IP .EQ. 2) WRITE(LU,102)
0033        IF(IP .EQ. 3) WRITE(LU,103)
0034        IF(IP .EQ. 4) WRITE(LU,104)
0035        IF(IP .EQ. 5) WRITE(LU,105)
0036        IF(IP .EQ. 6) WRITE(LU,106)
0037        IF(IP .EQ. 7) WRITE(LU,107)
0038        IF(IP .EQ. 8) WRITE(LU,108)
0039        CALL SETAR(IGCB,1.5)
0040        CALL VIEWP(IGCB,VXMIN,VXMAX,VYMIN,VYMAX)
0041        CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
0042        HGT=3.0
0043        AR=0.4
0044        SANG=0.0
0045        ICOOR=0
0046        CALL CSIZE(IGCB,HGT,AR,SANG,ICOOR)
0047        CALL FXD(IGCB,0)
0048        XTIC=(XMIN-XMAX)/5.
0049        YTIC=(YMAX-YMIN)/5.
0050        IF(IFRAM.NE.1) GO TO 42
0051        CALL FRAME(IGCB)
0052        CALL FRAME(IGCB)
0053        CALL LAXES(IGCB,XTIC,YTIC,XMIN,YMIN,1.,1.,2.)
0054        VXMI3=VXMIN-3.
0055        VXMA3=VXMAX+6.
0056        VYMI3=VYMIN-3.
0057        VYMA3=VYMAX+3.
0058        DO 43 IDIST=1,6
0059        DIST(IDIST)=(XMIN-(IDIST-1)*XTIC)*CALIB*.3/2.
0060     43 CONTINUE
0061        XCENT=(VXMAX+VXMIN)/2.
0062        YCENT=(VYMAX+VYMIN)/2.
0063        VXDIF=VXMIN-6.
0064        VYDIF=VYMIN-6.
```

```
0065          CALL VIEWP(IGCB,VXDIF,VXMAX,VYDIF,VYMAX)
0066          CALL WINDW(IGCB,VXDIF,VXMAX,VYDIF,VYMAX)
0067          CALL LORG(IGCB,1)
0068          CALL MOVE(IGCB,XCENT,VYDIF)
0069          CALL CPLOT(IGCB,-13.0,0.,0)
0070          CALL LABEL(IGCB)
0071          WRITE(LU,431) (DIST(IDIST),IDIST=1,6)
0072      431 FORMAT(1X,5(F4.1,6X),F4.1)
0073    C
0074          VXMA2=VXMA3-3.
0075          CALL MOVE(IGCB,VXMA2,VYMIN)
0076          CALL LDIR(IGCB,1.57)
0077          IF(IBUF1(8).EQ.2) GO TO 30
0078          CALL LABEL(IGCB)
0079          WRITE(LU,432) NAVG,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0080          GO TO 31
0081       30 CONTINUE
0082          CALL LABEL(IGCB)
0083          WRITE(LU,433) NAVG,MON,IDAY,IYEAR,IHOUR,MINUT,IAM
0084       31 CONTINUE
0085          CALL LDIR(IGCB,0.)
0086      432 FORMAT(I4," SHOT AVERAGE - HORIZONTAL",/,
0087         -6X,I2,"/",I2,"/",I2,4X,I2,":",I2,1X,A2)
0088      433 FORMAT(I4," SHOT AVERAGE -  VERTICAL ",/,
0089         -6X,I2,"/",I2,"/",I2,4X,I2,":",I2,1X,A2)
0090          CALL MOVE(IGCB,VXMIN,VYMIN)
0091          CALL VIEWP(IGCB,VXMIN,VXMAX,VYMIN,VYMAX)
0092          CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
0093    C
0094    C
0095       42 CONTINUE
0096          DO 20 J=3,NPTS
0097          IF(J.GT.3) GO TO 21
0098          IF(ICRV.NE.1)CALL MOVE(IGCB,X(J),BUFSU(J))
0099          IF(ICRV.EQ.1)CALL MOVE(IGCB,X(J),BUFSU(J))
0100          GO TO 20
0101       21 IF(ICRV.NE.1)CALL DRAW(IGCB,X(J),BUFSU(J))
0102          IF(ICRV.EQ.1)CALL DRAW(IGCB,X(J),BUFSU(J))
0103       20 CONTINUE
0104          XTIME=XMIN+3.2*(XMAX-XMIN)/5.
0105          YTIME=YMAX-ICRV*(YMAX-YMIN)/10.
0106          CALL MOVE(IGCB,XTIME,YTIME)
0107          CALL LABEL(IGCB)
0108          IF(ICRV.EQ.1) WRITE(LU,201)
0109    C     IF(ICRV.EQ.2) WRITE(LU,202)
0110          IF(ICRV.EQ.2) WRITE(LU,203)
0111      201 FORMAT(" AVE. NO FIT")
0112    C 202 FORMAT(" AVE. 3-PTS. FIT")
0113      203 FORMAT(" AVE. 5-PTS. FIT")
0114          CALL XMIT(IGCB)
0115       41 CONTINUE
0116          IF(NPLT.EQ.NPLOT.AND.ICRV.EQ.NCRV) WRITE(LU,100)
0117          IF(NPLT.EQ.NPLOT.AND.ICRV.EQ.NCRV) CALL PLOTR(IGCB,ID,0)
0118          IF(ICRV.EQ.1) GO TO INSTN3
0119          IF(ICRV.NE.1) GO TO INSTN2
0120          END
0121          END$
```

```
&DPLOT T=00004 IS ON CR00012 USING 00018 BLKS R=0000
```



```
0001   FTN,L
0002         PROGRAM DPLOT
0003         COMMON/XNY/X(512),Y(512),NP
0004         COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0005        -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0006        -,NPLOT
0007         COMMON/CRED/XJUNK(20)
0008         COMMON INSTN1,INSTN2,INSTN3,INSTN4
0009         DIMENSION VPNOM(4,4)
0010         INTEGER REDUC(3),PTRED(3),DAFIT(3)
0011         EQUIVALENCE (IPRAM(1),LU),(IPRAM(2),ID)
0012         DATA VPNOM/10.,70.,57.,97.,10.,70.,7.,47.,85.,145.,57.,97.,
0013        -85.,145.,7.,47./
0014         DATA REDUC/2HRE,2HDU,2HC /
0015         DATA PTRED/2HPT,2HRE,2HD /
0016         DATA DAFIT/2HDA,2HFI,2H2 /
0017   C
0018   C  THIS CALL PICKS UP THE LU AND ID OF THE PLOTTING DEVICE
0019         CALL RMPAR(IPRAM)
0020   C
0021     600 FORMAT(" THIS PRORGAM IS THE MAIN PROGRAM FOR SETTING UP PLOTS",/,
0022        -       " IT READS IN THE PLOT DEVICE LU AND ID THRU RMPAR....",/,
0023        -       " I.E. RU,GPLOT,1,1    PLOTS ON THE CRT  OR",/,
0024        -       "      RU,GPLOT,32,2   PLOTS ON THE PLOTTER",/)
0025     601 FORMAT(" QUESTIONS WILL BE ASKED CONCERNING HOW MANY PLOTS",/,
0026        -       " SCALING OF EACH (INDIVIDUALLY) AND PEN NUMBER",/)
0027         WRITE(1,600)
0028         WRITE(1,601)
0029   C
0030   C  THIS ASSURES THAT A VALID LU/ID COMBINATION IS USED
0031         IF(LU.LE.0 .OR. ID.LE.0) GO TO 90
0032         IF(ID.EQ.1) LU=1
0033         IF(ID.EQ.2.AND.(LU.NE.32.AND.LU.NE.33)) GO TO 90
0034         IF(ID.GT.2) GO TO 90
0035   C
0036     300 FORMAT(1X,"HOW MANY FRAMES DO YOU WANT?")
0037   C
0038     400 FORMAT(" INPUT VIEWPORT COORD.: VXMIN,VXMAX,VYMIN,VYMAX",/,
0039        -" FOR FRAME",I3," DEFAULT VALUES:",3(F3.0,","),F3.0)
0040     401 FORMAT(" INPUT WINDOW COORD. XMIN,XMAX,YMIN,YMAX",/,
0041        -" DEFAULT VALUES: 0,10,0,1000")
0042     402 FORMAT(" HOW MANY CURVES ON FRAME",I2,"?")
0043   C
0044     502 FORMAT(//" ***********************************************",/,
0045        -       " *              PLACE PEN IN HOLDER             *",/,
0046        -       " ***********************************************",//)
0047   C
0048         ASSIGN 101 TO INSTN1
0049         ASSIGN 102 TO INSTN2
0050         ASSIGN 103 TO INSTN3
0051         ASSIGN 104 TO INSTN4
0052   C
0053         WRITE(1,300)
0054         READ(1,*) NPLOT
0055         DO 92 NPLT=1,NPLOT
0056         VXMIN=0.
0057         VXMAX=0.
0058         KNOM=MOD(NPLT,4)
0059         IF(KNOM .EQ. 0) KNOM=4
0060         WRITE(1,400) NPLT,(VPNOM(JNOM,KNOM),JNOM=1,4)
0061         READ(1,*) VXMIN,VXMAX,VYMIN,VYMAX
0062         IF(VXMIN.NE.0. .OR. VXMAX.NE.0.) GO TO 30
0063         VXMIN=VPNOM(1,KNOM)
0064         VXMAX=VPNOM(2,KNOM)
```

```
0065          VYMIN=VPNOM(3,KNOM)
0066          VYMAX=VPNOM(4,KNOM)
0067       30 CONTINUE
0068    C
0069          XMIN=0.
0070          XMAX=0.
0071          WRITE(1,401)
0072          READ(1,*) XMIN,XMAX,YMIN,YMAX
0073          IF(XMIN.NE.0. .OR. XMAX.NE.0.) GO TO 31
0074          XMIN=0.
0075          XMAX=10.
0076          YMIN=0.
0077          YMAX=1000.
0078       31 CONTINUE
0079    C
0080    C     WRITE(1,402) NPLT
0081    C     READ(1,*) NCRV
0082    C     DO 92 ICRV=1,NCRV
0083          NCRV=2
0084          ICRV=1
0085          IF(LU.EQ.32) WRITE(1,502)
0086    C
0087          CALL EXEC(8,REDUC)
0088      101 CONTINUE
0089    C
0090          NP=XMAX/(.15*XJUNK(5))
0091          IF(NP.GT.500) NP=500
0092          CALL EXEC(8,PTRED)
0093      102 CONTINUE
0094    C
0095          CALL EXEC(8,DAFIT)
0096      103 CONTINUE
0097    C
0098          ICRV=2
0099          CALL EXEC(8,PTRED)
0100      104 CONTINUE
0101    C
0102       92 CONTINUE
0103          GO TO 91
0104       90 CONTINUE
0105          WRITE(1,999)
0106      999 FORMAT(1X,"IMPROPER LU OR ID...USE PARAMETERS IN RU COMMAND")
0107       91 CONTINUE
0108          END
0109          BLOCK DATA XYZ
0110          COMMON/XNY/X(512),Y(512),NP
0111          COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0112         -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0113         -,NPLOT
0114          COMMON/CRED/XJUNK(20)
0115          COMMON INSTN1,INSTN2,INSTN3,INSTN4
0116          END
0117          END$
```

```
&REDUC T=00004 IS ON CR00012 USING 00024 BLKS R=0000
```



```
0001   FTN4,L
0002         PROGRAM REDUC(5)
0003         COMMON /XNY/ X(512),Y(512),NP
0004         COMMON /CRED/ XJUNK(20)
0005         COMMON INSTN1,INSTN2,INSTN3,INSTN4
0006         DIMENSION IBUFF1(15),IBUFF2(514),SUM(512)
0007         DIMENSION IDAY(2),IMON(2),IYEAR(2),IHOUR(2),IMIN(2)
0008   C
0009     100 FORMAT(20X," SHOTS  H/V  T/P OP/BL")
0010     101 FORMAT(2X,I2,"/",I2,"/",I2,2X,I2,":",I2,1X,A2,3I5,3X,A2,5I5)
0011     102 FORMAT(2X,2I5)
0012     103 FORMAT("  SKIPPING .....")
0013   C
0014     200 FORMAT(" HOW MANY FILES SHOULD I SKIP?")
0015     201 FORMAT(" HOW MANY FILES HAVE TO BE READ TO OBTAIN THOSE FILES?")
0016     202 FORMAT(" SHOULD I REWIND THE TAPE?  (Y OR N)")
0017     203 FORMAT(A1)
0018     204 FORMAT(" HOW MANY FILES SHOULD I BACKSPACE?")
0019     205 FORMAT(" INPUT THE LU FOR THE OUTPUT")
0020     206 FORMAT(" IT IS ASSUMED THAT THE SKIP AND BACKSPACE COMMANDS",/,
0021        -       " POSITIONED THE TAPE CLOSE TO THE DESIRED DATA.  NOW",/)
0022     207 FORMAT(/"  INPUT THE TIME FOR THE BLOCKED CASE IN THE FORM",/,
0023        -         " 9,45")
0024     208 FORMAT(/"  INPUT THE TIME FOR THE   OPEN  CASE IN THE FORM",/,
0025        -         " 9,46")
0026     209 FORMAT(/" INPUT THE TYPE OF SMOOTHING THAT YOU WANT DONE",/,
0027        -         "  0 - NO SMOOTHING",/,"  1 - RECTANGULAR SMOOTHING",/,
0028        -         "  2 - TRIANGULAR SMOOTHING (DIST WILL BE FULL WIDTH AT"
0029        -      ," HALF MAX)",/,"  3 - GAUSSIAN (DIST IS 2X 1-SIGMA WIDTH)")
0030     210 FORMAT(/" INPUT 'DIST', THE SIZE IN KILOMETERS OF THE SMOOTHING"
0031        -        ," FILTER.")
0032   C
0033         ICNWD=10B
0034         IREWND=41OB
0035         ICBK=1410B
0036         ICFWD=1310B
0037         IYES=1HY
0038         NO=1HN
0039   C
0040         WRITE(1,202)
0041         READ(1,203) IANS
0042         IF(IANS.EQ.IYES) CALL EXEC(3,IREWND)
0043   C
0044         WRITE(1,200)
0045         READ(1,*) NSKIP
0046         IF(NSKIP.EQ.0) GO TO 11
0047         WRITE(1,103)
0048         WRITE(1,100)
0049         DO 10 I=1,NSKIP
0050         CALL EXEC(1,ICNWD,IBUFF1,15)
0051         WRITE(1,101) (IBUFF1(J),J=1,15)
0052         IF(IBUFF1(1).EQ.13 ) GO TO 99
0053         CALL EXEC(3,ICFWD)
0054      10 CONTINUE
0055      11 CONTINUE
0056   C
0057         WRITE(1,204)
0058         READ(1,*) NBACK
0059         IF(NBACK.EQ.0) GO TO 13
0060         NBACK1=NBACK+1
0061         DO 12 I=1,NBACK1
0062         CALL EXEC(3,ICBK)
0063      12 CONTINUE
0064         CALL EXEC(3,ICFWD)
```

```
0065      13 CONTINUE
0066   C
0067         DO 14 I=1,513
0068         X(I)=0.0
0069      14 CONTINUE
0070   C
0071         WRITE(1,206)
0072         WRITE(1,207)
0073         READ(1,*) IHOUR(1),IMIN(1)
0074         XJUNK(1)=IHOUR(1)
0075         XJUNK(2)=IMIN(1)
0076   C
0077         WRITE(1,208)
0078         READ(1,*) IHOUR(2),IMIN(2)
0079         XJUNK(3)=IHOUR(2)
0080         XJUNK(4)=IMIN(2)
0081   C
0082         WRITE(1,201)
0083         READ(1,*) NFILE
0084   C
0085         WRITE(1,100)
0086         DO 30 K=1,NFILE
0087         CALL EXEC(1,ICNWD,IBUFF1,15)
0088         WRITE(1,101) (IBUFF1(I),I=1,15)
0089         IF(IBUFF1(1).EQ.13 .AND. IBUFF1(4).EQ.13) GO TO 99
0090         IF(IBUFF1(4).EQ.IHOUR(2) .AND. IBUFF1(5).EQ.IMIN(2)) GO TO 16
0091         IF(IBUFF1(4).EQ.IHOUR(1) .AND. IBUFF1(5).EQ.IMIN(1)) GO TO 16
0092         GO TO 24
0093      16 CONTINUE
0094         NAVG=IBUFF1(7)
0095         XJUNK(17)=NAVG
0096         DO 15 I=1,512
0097         SUM(I)=0.0
0098      15 CONTINUE
0099   C
0100         DO 20 J=1,NAVG
0101         CALL EXEC(1,ICNWD,IBUFF2,514)
0102         DO 18 I=3,514
0103         SUM(I-2)=SUM(I-2)+IBUFF2(I)
0104      18 CONTINUE
0105      20 CONTINUE
0106   C
0107         YAVG=1.0/FLOAT(NAVG)
0108         IF(IBUFF1(4).NE.IHOUR(2) .OR. IBUFF1(5).NE. IMIN(2)) GO TO 22
0109         IDAY(2)=IBUFF1(1)
0110         IMON(2)=IBUFF1(2)
0111         IYEAR(2)=IBUFF1(3)
0112         XJUNK(6)=IDAY(2)
0113         XJUNK(7)=IMON(2)
0114         XJUNK(8)=IYEAR(2)
0115         XJUNK(9)=IBUFF1(6)
0116         XJUNK(16)=IBUFF1(8)
0117         CALIB=FLOAT(IBUFF1(9))/10.
0118         CALL VSMY(YAVG,SUM,1,Y,1,512)
0119      22 CONTINUE
0120         IF(IBUFF1(4).NE.IHOUR(1) .OR. IBUFF1(5).NE.IMIN(1)) GO TO 24
0121         IDAY(1)=IBUFF1(1)
0122         IMON(1)=IBUFF1(2)
0123         IYEAR(1)=IBUFF1(3)
0124         CALL VSMY(YAVG,SUM,1,X,1,512)
0125      24 CONTINUE
0126   C
0127         CALL EXEC(3,ICFWD)
0128   C
0129      30 CONTINUE
0130         WRITE(1,209)
```

```
0131          READ(1,*) IREC
0132          XJUNK(18)=IREC
0133          WRITE(1,210)
0134          READ(1,*) DIST
0135          IF(IREC.NE.0) CALL SMOTH(Y,DIST,CALIB,IREC)
0136          IF(IREC.NE.0) CALL SMOTH(X,DIST,CALIB,IREC)
0137          XJUNK(19)=DIST
0138          CALL VSUB(Y,1,X,1,Y,1,512)
0139          XJUNK(5)=CALIB
0140          DO 40 I=1,512
0141          X(I)=I*CALIB*.15
0142       40 CONTINUE
0143    C
0144          GO TO 999
0145       99 CONTINUE
0146          WRITE(1,900)
0147          CALL EXEC(3,ICBK)
0148          CALL EXEC(3,ICFWD)
0149      900 FORMAT(" FILE READ TERMINATED *** TRASH FILE ENCOUNTERED")
0150      999 CONTINUE
0151          GO TO INSTN1
0152          END
0153          SUBROUTINE SMOTH(Y,DIST,CALIB,IREC)
0154          DIMENSION Y(512),T(512),W(801),WSUM(512)
0155    C  IREC=1  RECTANGULAR SMOOTHING
0156    C  IREC=2  TRIANGULAR SMOOTHING
0157    C  IREC=3  GAUSSIAN SMOOTHING
0158          NP=DIST/(CALIB*.15)
0159          IF(IREC.NE.1) NP=2.0*DIST/(CALIB*.15)
0160          IF(IREC.EQ.3) NSIG=0.5*DIST/(CALIB*.15)
0161          IF(IREC.EQ.3) MU=2*NSIG
0162          IF(IREC.EQ.3) NP=2*MU+1
0163          IF(NP.GT.800) NP=800
0164          NP2=NP/2
0165          NP=NP2*2+1
0166          DIST=NP*CALIB*.15
0167          IF(IREC.NE.1) DIST=NP*CALIB*.15/2.0
0168    C     IF(IREC.EQ.3) DIST=NP*CALIB*.15/2.0
0169    C
0170          DO 10 I=1,NP2
0171          IF(IREC.EQ.1) W(I)=1.
0172          IF(IREC.EQ.2) W(I)=FLOAT(I)/FLOAT(NP2+1)
0173          IF(IREC.EQ.3) W(I)=EXP(-.5*(FLOAT(I-MU-1)/FLOAT(NSIG))**2)
0174          W(NP+1-I)=W(I)
0175       10 CONTINUE
0176          W(NP2+1)=1.0
0177          DO 15 I=1,512
0178          MIN=I-NP2
0179          MAX=I+NP2
0180          IF(MIN.LT.1) MIN=1
0181          IF(MAX.GT.512) MAX=512
0182          T(I)=0.0
0183          WSUM(I)=0.0
0184          DO 15 J=MIN,MAX
0185          INDEX=J-I+NP2+1
0186          T(I)=T(I)+W(J-I+NP2+1)*Y(J)
0187          WSUM(I)=WSUM(I)+W(J-I+NP2+1)
0188       15 CONTINUE
0189          DO 20 I=1,512
0190          Y(I)=T(I)/WSUM(I)
0191       20 CONTINUE
0192          RETURN
0193          END
0194          END$
```

```
&DAFI2 T=00004 IS ON CR00012 USING 00012 BLKS K=0000
```



```
0001   FTN4,L
0002         PROGRAM DAFI2(5)
0003         COMMON /XNY/ X(512),S(512),NP
0004         COMMON /CRED/ XJUNK(20)
0005         COMMON INSTN1,INSTN2,INSTN3,INSTN4
0006         DIMENSION PARA(4),PSTR(3,4),SIG(3),SIGT(4),TEMP(4),IC(4),JIN(4)
0007   C
0008     200 FORMAT(" INPUT THE MIN DIST. AND MAX DIST. FOR THE FIT. EG 0,6")
0009     201 FORMAT(" INPUT THE ORDER IN WHICH YOU WANT TO FIT. EG. 2,1,3,4",
0010        -/,  " WILL FIT AREA, KAPPA, ALPHA, FOCUS.")
0011         CALIB=XJUNK(5)
0012         WRITE(1,200)
0013         READ(1,*) RMIN,RMAX
0014         N1=RMIN/(.15*CALIB)
0015         N2=RMAX/(.15*CALIB)
0016         WRITE(1,201)
0017         READ(1,*) JIN(1),JIN(2),JIN(3),JIN(4)
0018   C
0019   C
0020         RCEN=(RMAX+RMIN)/2.
0021         I1=RCEN/(.15*CALIB)
0022         I2=RMIN/(.15*CALIB)
0023         I3=RMAX/(.15*CALIB)
0024         G1=(1.0-RCEN/2.)**2
0025         G2=(1.0-RMIN/2.)**2
0026         G3=(1.0-RMAX/2.)**2
0027         T1=EXP(-.3*RCEN)/SQRT((RCEN*1000.)**2+(.040*1.0E+05/1.06)**2*G1)
0028         T2=EXP(-.3*RMIN)/SQRT((RMIN*1000.)**2+(.040*1.0E+05/1.06)**2*G2)
0029         T3=EXP(-.3*RMAX)/SQRT((RMAX*1000.)**2+(.040*1.0E+05/1.06)**2*G3)
0030         PSTR(1,1)=S(I1)/T1
0031         PSTR(2,1)=S(I2)/T2
0032         PSTR(3,1)=S(I3)/T3
0033   C
0034         PSTR(1,2)=0.04
0035         PSTR(2,2)=0.025
0036         PSTR(3,2)=0.055
0037   C
0038         PSTR(1,3)=0.3
0039         PSTR(2,3)=0.1
0040         PSTR(3,3)=0.5
0041   C
0042         PSTR(1,4)=2.
0043         PSTR(2,4)= .5
0044         PSTR(3,4)= 5.
0045   C
0046         DO 99 N=1,10
0047         DO 13 J=1,4
0048         IF(J.EQ.1) KJ=JIN(1)
0049         IF(J.EQ.2) KJ=JIN(2)
0050         IF(J.EQ.3) KJ=JIN(3)
0051         IF(J.EQ.4) KJ=JIN(4)
0052   C
0053   C  KJ = 1  KAPPA
0054   C       2  AREA
0055   C       3  ALPHA
0056   C       4  FOCAL RANGE
0057   C
0058         DO 12 I=1,3
0059   C
0060         DO 11 L=1,4
0061         PARA(L)=PSTR(1,L)
0062      11 CONTINUE
0063         PARA(KJ)=PSTR(I,KJ)
0064   C
```

```
0065          CALL SIGMA(PARA(1),PARA(2),PARA(3),PARA(4),SIG(I),S,N1,N2,X)
0066       12 CONTINUE
0067    C
0068          T1=(SIG(1)-SIG(2))/(PSTR(1,KJ)-PSTR(2,KJ))
0069          T2=(SIG(2)-SIG(3))/(PSTR(2,KJ)-PSTR(3,KJ))
0070          A2=(T1-T2)/(PSTR(1,KJ)-PSTR(3,KJ))
0071          A1=T2-A2*(PSTR(2,KJ)+PSTR(3,KJ))
0072          A0=SIG(3)-A1*PSTR(3,KJ)-A2*PSTR(3,KJ)**2
0073          UPDAT=-A1/(2.*A2)
0074    C
0075          IMAX=1
0076          IF(SIG(2).GT.SIG(1)) IMAX=2
0077          IF(SIG(3).GT.SIG(IMAX)) IMAX=3
0078          IMIN=1
0079          IF(SIG(2).LT.SIG(1)) IMIN=2
0080          IF(SIG(3).LT.SIG(IMIN)) IMIN=3
0081          ICEN=1
0082          IF(ICEN.EQ.IMAX .OR. ICEN.EQ.IMIN) ICEN=2
0083          IF(ICEN.EQ.IMAX .OR. ICEN.EQ.IMIN) ICEN=3
0084          SLOPE=(PSTR(IMIN,KJ)-PSTR(ICEN,KJ))/(SIG(IMIN)-SIG(ICEN))
0085          PARA(KJ)=PSTR(IMIN,KJ)-SLOPE*SIG(IMIN)
0086          IF(PARA(KJ).LT.0.) PARA(KJ)=ABS(PARA(KJ))
0087          IFLAG=0
0088          IF(UPDAT.LT.PSTR(IMAX,KJ).AND.UPDAT.GT.PSTR(IMIN,KJ).AND.A2.GT.0.)
0089         -     IFLAG=1
0090          IF(UPDAT.LT.PSTR(IMAX,KJ).AND.UPDAT.GT.PSTR(IMIN,KJ).AND.A2.GT.0.)
0091         -    PARA(KJ)=UPDAT
0092          IF2=1
0093    C     WRITE(31,101) PARA(KJ),PSTR(IMIN,KJ),SIG(IMIN),PSTR(ICEN,KJ),
0094    C    -                SIG(ICEN),PSTR(IMAX,KJ),SIG(IMAX)
0095      101 FORMAT(E12.3,6E9.3)
0096          IF(PARA(KJ).GT.10.*PSTR(IMIN,KJ)) PARA(KJ)=PSTR(IMIN,KJ)
0097          DO 14 I=1,10
0098          IF(I.GT.1) PARA(KJ)=(PSTR(IMIN,KJ)+PARA(KJ))/2.
0099          CALL SIGMA(PARA(1),PARA(2),PARA(3),PARA(4),SDEV,S,N1,N2,X)
0100          IF(SDEV.LT.SIG(IMIN)) GO TO 15
0101       14 CONTINUE
0102          IF2=0
0103       15 CONTINUE
0104          TEMP(1)=PSTR(1,KJ)
0105          TEMP(2)=PSTR(2,KJ)
0106          TEMP(3)=PSTR(3,KJ)
0107          TEMP(4)=PARA(KJ)
0108          SIGT(1)=SIG(1)
0109          SIGT(2)=SIG(2)
0110          SIGT(3)=SIG(3)
0111          SIGT(4)=SDEV
0112          YSTOR=SIGT(1)
0113          IC(1)=1
0114          DO 21 I=1,4
0115          DO 20 L=I,4
0116          IF(SIGT(L).GT.YSTOR) GO TO 20
0117          YSTOR=SIGT(I)
0118          IC(I)=L
0119       20 CONTINUE
0120          INDX=IC(I)
0121          SIGT(INDX)=SIGT(I)
0122          SIGT(I)=YSTOR
0123          PSTOR=TEMP(INDX)
0124          TEMP(INDX)=TEMP(I)
0125          TEMP(I)=PSTOR
0126       21 CONTINUE
0127    C
0128          PSTR(1,KJ)=TEMP(1)
0129          PSTR(2,KJ)=TEMP(2)
0130          PSTR(3,KJ)=TEMP(3)
```

```
0131      16 CONTINUE
0132  C
0133         WRITE(1,100) SIGT(1),(PSTR(1,I),I=1,4),IFLAG,IF2
0134     100 FORMAT(" SDEV,XK,A,AL,F=",5E9.3,2I2)
0135  C
0136      13 CONTINUE
0137  C
0138      99 CONTINUE
0139         XJUNK(10)=PSTR(1,3)
0140         XJUNK(11)=PSTR(1,2)
0141         XJUNK(12)=PSTR(1,4)
0142         XJUNK(13)=PSTR(1,1)
0143         XJUNK(14)=RMIN
0144         XJUNK(15)=RMAX
0145  C
0146         CALL CALC(PSTR(1,1),PSTR(1,2),PSTR(1,3),PSTR(1,4),X,S)
0147         GO TO INSTN3
0148         END
0149  C
0150  C
0151         SUBROUTINE SIGMA(XK,A,AL,F,SDEV,S,N1,N2,X)
0152         DIMENSION S(512),X(512)
0153         SUM=0.
0154         SUM2=0.
0155         DO 10 I=N1,N2
0156         T1=XK*EXP(-AL*X(I))
0157         T2=(1.0-X(I)/F)**2
0158         T2=T2*1.0E+10*A*A/(1.06*1.06)
0159         T2=(X(I)*1000.)**2+T2
0160         SCALC=T1/SQRT(T2)
0161         SUM=SUM+(S(I)-SCALC)**2
0162         SUM2=SUM2+(S(I)-SCALC)
0163      10 CONTINUE
0164         SDEV=SQRT(SUM/FLOAT(N2-N1+1))
0165         SUM2=SUM2/FLOAT(N2-N1+1)
0166  C      WRITE(31,100) SDEV,SUM2,XK,A,AL,F
0167     100 FORMAT(" SDEV,SUM2,XK,A,AL,F=",6E9.3)
0168  C
0169         RETURN
0170         END
0171  C
0172  C
0173         SUBROUTINE CALC(XK,A,AL,F,X,Y)
0174         DIMENSION X(512),Y(512)
0175         DO 10 I=1,512
0176         T1=XK*EXP(-AL*X(I))
0177         T2=(1.0-X(I)/F)**2
0178         T2=T2*A*A*1.0E+10/(1.06*1.06)
0179         T2=T2+(X(I)*1000.)**2
0180         Y(I)=T1/SQRT(T2)
0181      10 CONTINUE
0182         RETURN
0183         END
0184         END$
```

```
&PTRED T=00004 IS ON CR00012 USING 00018 BLKS R=0000

0001  FTN,L
0002        PROGRAM PTRED(5)
0003        COMMON/XNY/X(512),Y(512),NP
0004        COMMON/GRAPH/IGCB(192),IPRAM(5),IOBUF(20),IDCB(144),IFILE(3)
0005       -,XMIN,XMAX,YMIN,YMAX,NCRV,ICRV,VXMIN,VXMAX,VYMIN,VYMAX,NPLT
0006       -,NPLOT
0007        COMMON/CRED/XJUNK(20)
0008        COMMON INSTN1,INSTN2,INSTN3,INSTN4
0009    100 FORMAT("SP0")
0010    101 FORMAT("SP1")
0011    102 FORMAT("SP2")
0012    103 FORMAT("SP3")
0013    104 FORMAT("SP4")
0014    105 FORMAT("SP5")
0015    106 FORMAT("SP6")
0016    107 FORMAT("SP7")
0017    108 FORMAT("SP8")
0018        LU=IPRAM(1)
0019        ID=IPRAM(2)
0020        IF(ID.EQ.1.AND.NPLT.EQ.1.AND.ICRV.EQ.1)
0021       -CALL PLOTR(IGCB,ID,4,LU,IOBUF,20)
0022        IF(ID.EQ.2.AND.NPLT.EQ.1.AND.ICRV.EQ.1)
0023       -CALL PLOTR(IGCB,ID,4,LU,IOBUF,20)
0024        IP=MOD(ICRV,8)
0025        IF(IP.LE.0) IP=8
0026        IF(IP .EQ. 1) WRITE(LU,101)
0027        IF(IP .EQ. 2) WRITE(LU,102)
0028        IF(IP .EQ. 3) WRITE(LU,103)
0029        IF(IP .EQ. 4) WRITE(LU,104)
0030        IF(IP .EQ. 5) WRITE(LU,105)
0031        IF(IP .EQ. 6) WRITE(LU,106)
0032        IF(IP .EQ. 7) WRITE(LU,107)
0033        IF(IP .EQ. 8) WRITE(LU,108)
0034  C     IF(IP .EQ. 1.AND.LU.NE.32) WRITE(LU,101)
0035  C     IF(IP .EQ. 2.AND.LU.NE.32) WRITE(LU,102)
0036  C     IF(IP .EQ. 3.AND.LU.NE.32) WRITE(LU,103)
0037  C     IF(IP .EQ. 4.AND.LU.NE.32) WRITE(LU,104)
0038  C     IF(IP .EQ. 5.AND.LU.NE.32) WRITE(LU,105)
0039  C     IF(IP .EQ. 6.AND.LU.NE.32) WRITE(LU,106)
0040  C     IF(IP .EQ. 7.AND.LU.NE.32) WRITE(LU,107)
0041  C     IF(IP .EQ. 8.AND.LU.NE.32) WRITE(LU,108)
0042        CALL SETAR(IGCB,1.5)
0043        CALL VIEWP(IGCB,VXMIN,VXMAX,VYMIN,VYMAX)
0044        CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
0045        HGT=3.0
0046        AR=0.4
0047        SANG=0.0
0048        ICOOR=0
0049        CALL CSIZE(IGCB,HGT,AR,SANG,ICOOR)
0050        CALL FXD(IGCB,0)
0051        XTIC=(XMIN-XMAX)/5.
0052        YTIC=(YMAX-YMIN)/5.
0053        IF(IP.EQ.1) CALL FRAME(IGCB)
0054        IF(IP.EQ.1) CALL FRAME(IGCB)
0055        IF(IP.EQ.1)CALL LAXES(IGCB,XTIC,YTIC,XMIN,YMIN,1.,1.,2.)
0056        DO 20 J=1,NP
0057        IF(J.GT.1) GO TO 21
0058        CALL MOVE(IGCB,X(J),Y(J))
0059        GO TO 20
0060     21 CALL DRAW(IGCB,X(J),Y(J))
0061     20 CONTINUE
0062        IF(ICRV.GT.1) GO TO 10
0063        VXMI3=VXMIN-3.
0064        VXMA3=VXMAX+6.
```

```
0065          VYMI3=VYMIN-3.
0066          VYMA3=VYMAX+3.
0067          CALL VIEWP(IGCB,VXMI3,VXMA3,VYMI3,VYMA3)
0068          CALL WINDW(IGCB,VXMI3,VXMA3,VYMI3,VYMA3)
0069          CALL LORG(IGCB,1)
0070          VXMA2=VXMA3-3.
0071          DIST=XJUNK(19)
0072          NAVG=XJUNK(17)
0073          IM=XJUNK(6)
0074          IDAY=XJUNK(7)
0075          IYER=XJUNK(8)
0076          IREC=XJUNK(18)
0077          ILOOK=XJUNK(16)
0078          CALL MOVE(IGCB,VXMA2,VYMIN)
0079          CALL LDIR(IGCB,1.57)
0080          IF(ILOOK.EQ.2)GO TO 30
0081          CALL LABEL(IGCB)
0082          WRITE(LU,432)NAVG,IM,IDAY,IYER
0083      432 FORMAT(I4," SHOTS-AVERAGE  HORIZONTAL",/,
0084         -12X,I2,"/",I2,"/",I2)
0085          GO TO 31
0086       30 CONTINUE
0087          CALL LABEL(IGCB)
0088          WRITE(LU,433)NAVG,IM,IDAY,IYER
0089      433 FORMAT(I4,"  SHOTS-AVERAGE  VERTICAL",/,
0090         -12X,I2,"/",I2,"/",I2)
0091       31 CONTINUE
0092          CALL LDIR(IGCB,0.)
0093          XCENT=(VXMAX+VXMIN)/2.
0094          YCENT=(VYMAX+VYMIN)/2.
0095          VXDIF=VXMIN-2.*HGT
0096          VYDIF=VYMIN-2.*HGT
0097          CALL VIEWP(IGCB,VXDIF,VXMAX,VYDIF,VYMAX)
0098          CALL WINDW(IGCB,VXDIF,VXMAX,VYDIF,VYMAX)
0099          CALL LORG(IGCB,1)
0100          CALL MOVE(IGCB,XCENT,VYDIF)
0101          CALL CPLOT(IGCB,-13.,0.,0)
0102    C
0103          CALL LABEL(IGCB)
0104          WRITE(LU,700)
0105    C
0106          CALL MOVE(IGCB,VXDIF,YCENT)
0107          CALL CPLOT(IGCB,0.,-5.,0)
0108          CALL LDIR(IGCB,1.57)
0109          CALL LABEL(IGCB)
0110          WRITE(LU,701)
0111          CALL LDIR(IGCB,0.)
0112    C
0113          CALL VIEWP(IGCB,VXMIN,VXMAX,VYMIN,VYMAX)
0114          CALL WINDW(IGCB,XMIN,XMAX,YMIN,YMAX)
0115    C
0116          XTIME=XMIN+3.2*(XMAX-XMIN)/5.
0117          YTIME=YMAX-(YMAX-YMIN)/10.
0118          CALL MOVE(IGCB,XTIME,YTIME)
0119          IH=XJUNK(1)
0120          IM=XJUNK(2)
0121          CALL LABEL(IGCB)
0122          WRITE(LU,702) IH,IM
0123          YTIME=YMAX-2.*(YMAX-YMIN)/10.
0124          CALL MOVE(IGCB,XTIME,YTIME)
0125          IH=XJUNK(3)
0126          IM=XJUNK(4)
0127          CALL LABEL(IGCB)
0128          WRITE(LU,703) IH,IM
0129          XTIME=XMIN+2.6*(XMAX-XMIN)/5.
0130          YTIME=YMAX-3.*(YMAX-YMIN)/10.
```

```
0131          CALL MOVE(IGCB,XTIME,YTIME)
0132          GO TO (505,506,507,508),(IREC+1)
0133      505 CONTINUE
0134          CALL LABEL(IGCB)
0135          WRITE(LU,705)
0136      705 FORMAT("       NO-SMOOTHING")
0137          GO TO 9
0138      506 CONTINUE
0139          CALL LABEL(IGCB)
0140          WRITE(LU,706)
0141      706 FORMAT("  RECTANGULAR-SMOOTHING")
0142          GO TO 9
0143      507 CONTINUE
0144          CALL LABEL(IGCB)
0145          WRITE(LU,707)
0146      707 FORMAT("   TRIANGULAR-SMOOTHING")
0147          GO TO 9
0148      508 CONTINUE
0149          CALL LABEL(IGCB)
0150          WRITE(LU,708)
0151      708 FORMAT("   GAUSSIAN-SMOOTHING")
0152        9 YTIME=YMAX-4.*(YMAX-YMIN)/10.
0153          XTIME=XMIN+2.7*(XMAX-XMIN)/5.
0154          CALL MOVE(IGCB,XTIME,YTIME)
0155          CALL LABEL(IGCB)
0156          WRITE(LU,709) DIST
0157      709 FORMAT(" FILTER WIDTH=",F4.2," KM.")
0158       10 CONTINUE
0159          IF(ICRV.EQ.1) GO TO 25
0160          RMAX=XJUNK(15)
0161          RMIN=XJUNK(14)
0162          AL=XJUNK(10)
0163          A=XJUNK(11)
0164          F=XJUNK(12)
0165          XK=XJUNK(13)
0166          XTIME=XMIN+3.2*(XMAX-XMIN)/5.0
0167          YTIME=YMAX-5.*(YMAX-YMIN)/10.
0168          CALL MOVE(IGCB,XTIME,YTIME)
0169          CALL LABEL(IGCB)
0170          WRITE(LU,704) AL,A,F,XK
0171          XG01=XMIN+RMIN
0172          XG02=XMIN+RMAX
0173          YG01=YMAX-(YMAX-YMIN)/10.
0174          CALL MOVE(IGCB,XG01,YMAX)
0175          CALL PLOT(IGCB,XG01,YMAX,-2)
0176          CALL PLOT(IGCB,XG01,YG01,-1)
0177          CALL MOVE(IGCB,XG02,YMAX)
0178          CALL PLOT(IGCB,XG02,YMAX,-2)
0179          CALL PLOT(IGCB,XG02,YG01,-1)
0180       25 CONTINUE
0181      700 FORMAT("   DISTANCE, KILOMETERS     ")
0182      701 FORMAT("RELATIVE SIGNAL, ARB. UNITS")
0183      702 FORMAT("BLOCKED",I3,":",I2)
0184      703 FORMAT("  OPEN ",I3,":",I2)
0185      704 FORMAT("AL=",E10.3,/,"A =",E10.3,/,"F =",E10.3,/,"XK=",E10.3)
0186          CALL XMIT(IGCB)
0187          IF(NPLT.EQ.NPLOT.AND.ICRV.EQ.NCRV) WRITE(LU,100)
0188          IF(NPLT.EQ.NPLOT.AND.ICRV.EQ.NCRV) CALL PLOTR(IGCB,ID,0)
0189          IF(ICRV.EQ.1) GO TO INSTN2
0190          IF(ICRV.EQ.2) GO TO INSTN4
0191          END
0192          END$
```

ORIGINAL PAGE IS
OF POOR QUALITY