NASA CR-152595

N77-33742

TORNADO DETECTION DATA REDUCTION AND ANALYSIS

FINAL REPORT

NASA Contract NAS5-23508

(NASA-CR-152595) TORNADO DETECTION DATA REDUCTION AND ANALYSIS Final Report, Jul. 1976 - Jul. 1977 (Techno-Sciences, Inc., Annapolis, Md.) 23 p HC A02/MF A01 CSCL 04B

N77-33742

Unclas

G3/47 49961

TECHNO-SCIENCES INC

1712 SOUTH HARBOR LANE
ANNAPOLIS, MARYLAND
21401

TSI Report No. 77163
August 1977

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, Maryland 20771

TECHNICAL REPORT STANDARD TITLE PAGE

N77-33742

1 Report No 77163	2 Government Accession No	3 Recipient's Catalog No
4 Title and Subtitle TORNADO DETECTION DATA REDUCTION AND ANALYSIS		5. Report Date August 1977
		6 Performing Organization Code
7 Author(s) L. D. Davisson		8 Performing Organization Report No 77163
9 Performing Organization Name and Address Techno-Sciences, Inc 1712 S. Harbor Lane Annapolis, Maryland 21401		10. Work Unit No.
		11 Contract or Grant No NAS5-23508
12 Sponsoring Agency Name and Address National Aeronautics and Space Administration Goddard Space Flight Center Greenblet, Maryland 20771		13. Type of Report and Period Covered Type III Final July 1976-July 1977
		14 Sponsoring Agency Code
15 Supplementary Notes		

16 Abstract

This report reviews work performed under NASA Contract NAS5-23508. The work consisted of the development of various data reduction and analysis computer programs with applications to measured data. A computer simulation was also developed.

17 Key Words Suggested by Author	18 Distribution Statement

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price

Page
Intentionally
Left Blank

PREFACE

The purpose of this contract was to provide data processing and analysis in support of two GSFC programs: tornado detection by analysis of radio frequency interference in various frequency bands and sea state determination from short pulse radar measurements. Extensive data reduction efforts were performed in support of these programs. A new and unique backscatter simulation was implemented to predict radar performance as a function of the wind velocity. Computer programs were developed for the various data processing and analysis goals of the contract effort and delivered to GSFC.

I. Introduction

The data supplied to Techno-Sciences for reduction and analysis consist of 9 track, 800 bpi digital tapes in various record formats and include, in addition to the data records, records indicating the wind state at the measuring platform, aircraft orientation, altitude, and time of recording. These data were collected over several experiments and have several corresponding formats. Therefore it was necessary for Techno-Sciences to develop highly flexible programs to be able to handle the various formats. Over 60 tapes have been involved in the analysis and data reduction job. These tapes were generated under a variety of sea conditions providing an effective data base for investigation.

An important tool in the difficult task of determining sea state conditions from short pulse radar data is the use of simulated data. Data generated by simulations is highly controlled so that it is possible to find and evaluate estimation methods without uncertainty as to the true value of the quantity to be estimated. Techno-Sciences devloped a 2 dimensional computer simulation during the contract period. The modelled data has a high degree of correlation with the measured data in appearance.

II. Short Pulse Radar Simulation

The problem of sea state estimation from measured short pulse radar information is very complex. Particularly complicating the problem is the lack of accurate "ground truth" data, i.e. the lack of exact knowledge of what an accurate estimate on any particular segment of data is. A good way of removing this problem is to develop computer simulations where the estimated parameters can be accurately known and controlled.

As a part of the contract effort, Techno-Sciences has developed a two-dimensional simulation of the received pulse under a Gaussian surface model. A listing of the program appears in Appendix A along with a sample printout of several pulses. The surface is allowed to consist of any combination of a deterministic sinusoidal component and a Gaussian component with a Pierce-Moskowitz spectrum under operator control. In each case, the period and amplitude is variable. The random portion is generated using an FFT on randomly generated frequency components whose variances are porportional to the Pierce-Moskowitz spectrum at each frequency. The simulation of the pulse return is based on equation (4) of Spectrum of Power Scattered by a Short Pulse From a Stochastic Surface, by David Levine, NASA report X-952-74-299, August 1974. The specular reflection points are found by interpolation between samples of the derivative of the surface where it goes through zero.

It has been found that the simulated results agree well with the measured data.

Appendix A

SHORT PULSE RADAR SIMULATION

Program Listing and Sample Printouts

Program SIMUL - Program Listing

```
$ASSM
SIMUL FRØG 2 D SHØFT PULSE SCATTER SIMULATIØN LDD
 EFLST
$FØRT
C FRØGFAM TØ SIMULATE A MØDIFICATIØN ØF EQN (4) IN THE TIME
C DØMAIN ØF "SPECTRUM ØF FØWER---" BY LEVINE ØF AUG 1974
C THE STATIØNAFY POINTS AFE FØUND FØR A GIVEN SURFACE
C USING FFT TECHNICUES
C EØTH A SINUSØIDAL AND A GAUSSIAN SURFACE
C CØMFØNENT ARE ALLØWED  THE SURFACE
C IS GENERATED IN THE FREQUENCY DØMAIN
C TØ GET THE DESIRED SFECTRUM  THE
C RANDOM SURFACE IS FIEF E-MØSKØWITZ
C WITH GIVEN PEAK FFEQUENCY AND
C RMS VALUE AS INFUT  THE SINUSØIDAL
C FART HAS THE SAME FERIØD BUT ANY
C ØTHEF CHØSEN RMMS VALUE
C FRØGRAM BY TECHNØ-SCIENCES,INC
C
C
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 IRAN
      CØMFLEX Z(1024),ZD(1024),ZDD(1024) CMPLX,CØNJG
      CØMFLEX W, ZS
      DIMENSION TEL(313),IGFD(21) IFLAT(21) LAST(31)
      DIMENSIØN FANGE(1024), FOOT(1024)
      ECUIVALENCE (FANGE(1),Z(1)) (CRØØT(1) ZDD(1))
      TAN(X)=SIN(X)/CØS(X)
C SFEC( )=FIERSØN-MØSKØWITZ SFECTRUM
      SFEC(X)=A2*E F(-AMIN.(130 .EG2U4/(X*X))  Y(**3)
C  =ANGULAR SURFACE FFEQUENCY
C IRAN=FANDØM NUMBEF GENERATOF
C GAUSSIAN PULSE SHAFE
      FULSE(X)=EXF(-((Y/SIGFLS)**2)/2  )
C (SIN( ) X)**2=ANTENNA FØWEFGAIN FATTERN
      GAIN(X)=(SIN(X/ANGVAL)/AMAX1(ABS(X/ANGVAL),1 E-3))**2
      DATA SIGTIN  OU6/
      DATA IRAN/1231321/
      DATA NFFT/1024
      DATA LU/3/
      DATA P2 S 2   .53/
      DATA C,DT FULSTM,FO/300   00.25 .01,13900 /
C C=SFEED OF LIGHT DT=TIME I F,FULSTM=FULSE DURATION
      DATA I3/3/
      DATA I4/4
C DTOR =DEGREES TO RADIANS CØNVEFSIØN
      DATA DTOR/ 01745329/
      DATA IØNE,IEIG/1,30000/
      DATA IGDFTS/ /
      WO=F0*F2
      NFFT2=NFFT 2
      NFFT21=NFFT2+1
```

```
      WRITE(0 25)
25    FØRMAT(59HENTEF RMS SINUSØIDAL(M ),
     1RMS RANDØM(F-M)(M ).PERIØD(M )(F8)   )
      FEAD(0,26) RMSSIN,RMSRAN,FEPIØD
26    FØRMAT(8F8)
      WRITE(0,27)
27    FØRMAT(33HENTEF RC 3DB CUTØFF*PULSE DUR(F8) )
      READ(0,26) FOTS
      WFITE(0,31)
31    FØRMAT(30HWAVE REPEATS.PULSE REPEATS(I5))
      FEAD(0,32) IRPTW,IRPTP
32    FØRMAT(8I5)
      WRITE(0,41)
41    FØRMAT(11HBLØCKS?(I5))
      FEAD(0,32) IELKS
      RHØ=EXP(-P2*FOTS*DT/PULSTM)
      PHØ1=SQRT(1 -RHØ*RHØ)
      BG2U4=F2*P2*1 5/(PERIØD*PERIØD)
      A2=P2*RMSRAN*RMSRAN*EG2U4
      ALTUD=2000.
      C2=C/2.
C CØNVERT FRØM PULSE TIME SIGMA TØ RANGE SIGMA
      SIGPLS=SIGTIM*C2
C RNGINC=ASSUMED PULSE STAFT.END POINT
      FNGINC=6 *SIGPLS
      ANGLIN=12
      ANGCTR=ANGLIN*DTØR
      BMWITH=25
      ANGSDE=BMWITH*DTØR/2
      ANGMAX=ANGCTF+ANGSDE
      ANGMIN=ANGCTF-ANGSDE
      ANGVAL=ANGSDE 1 39
      Y0=ALTUD*TAN(ANGMIN)
C SET UF GRID
      GDINC=1399 /FLØAT(IGDFTS-1)
      DØ 5 I=1,IGDFTS
5     IGRD(I)=FLØAT(I-1)*GDINC +.5
      YSDB=ALTUD*(TAN(ANGMAX)-TAN(ANGMIN))
      DY=YSDB FLØAT(NFFT)
      DY2=DY/2
      Y02=Y0-DY2
C DF*DY=1/NFFT
      DF=1 /YSDB
      FTDF=SQRT(DF)
      DW=P2*DF
C DW=DELTA ØMEGA,DF=DELTA FREQ
C SET UP FFT TAELE
      CALL FØUR1T(E NFFT,TEL,0)
C SET BLØCK CØUNTEF TØ ZERO BEFØRE LØØPING
      ICØUNT=0
C SET UF RANDØM FART OF WAVE TRANSFØRM WITH SFECTRUM SPEC(K)
C DØ FARTS=0
```

```
40      Z(1)=CMPLX(0.,0.)
        ZD(1)=CMPLX(0.,0.)
        ZDD(1)=CMPLX(0.,0.)
        DØ 1 I=2,NFFT21
        X=FLØAT(I-1)*DW
        Z(I)=CMPLX(SQRT(SPEC(X))*RANDG(IRAN),0.)*RTDF
1       CØNTINUE
        IFREQ=Y3DB/PERIØD+.5
C SET UP NØN-RANDØM WAVE CØMPØNENT
        Z(IFREQ)=Z(IFREQ)+CMPLX(RMSSIN/SQRT(2.),0.)
C SET UP FØR PLØT SCALE FACTØRS
        SCALE=RMSSIN**2
        X=FLØAT(IFREQ-1)*DW
        SCALE1=SCALE*X*X
        SCALE2=SCALE1*X*X
        DØ 2 I=2,NFFT21
        X=FLØAT(I-1)*DW
        XX=SPEC(X)*2.*DF
        SCALE=SCALE+XX
        XX=XX*X*X
        SCALE1=SCALE1+XX
        XX=XX*X*X
        SCALE2=SCALE2+XX
        W=CMPLX(0.,X)
        ZD(I)=W*Z(I)
        ZDD(I)=W*ZD(I)
2       CØNTINUE
        SCALE=SQRT(SCALE)
        SCALE1=SQRT(SCALE1)
        SCALE2=SQRT(SCALE2)
        DØ 3 I=2,NFFT21
        IC=NFFT+2-I
        Z(IC)=CONJG(Z(I))
        ZD(IC)=CØNJG(ZD(I))
        ZDD(IC)=CØNJG(ZDD(I))
3       CØNTINUE
        CALL FØUR1T(Z,NFFT,TEL,IØNE)
        CALL FØUR1T(ZD,NFFT,TEL,IØNE)
        CALL FØUR1T(ZDD,NFFT,TEL,IØNE)
C FIND STATIØNARY PØINTS ASSUMING LINEAR INTERPØLATIØN
C AND PLØT Z(Y), FIRST,2D DERIVATIVES
        CALL SETGRD(LU,IGRD,IGDPTS,IBIG)
        TMDIF=(YO2+DY)/ALTUD - REAL(ZD(1))
        K=0
        DØ 4 I=1,NFFT
        Y=FLØAT(I)*DY+YO2
        TANTHT=Y/ALTUD
        TANALF=REAL(ZD(I))
        TANDIF=TANTHT-TANALF
        IF (TANDIF*TMDIF.GT.0.) GØ TØ 6
        DLT=TMDIF/(TMDIF-TANDIF)
        DLT1=1.-DLT
```

```
      Y=Y+DLT*DY
      HEIGHT=REAL(Z(I-1))*DLT1+DLT*REAL(Z(I))
      DERIV=REAL(ZD(I-1))*DLT1+DLT*TANALF
      DERIV2=REAL(ZDD(I-1))*DLT1+DLT*REAL(ZDD(I))
      K=K+1
      RANGE(K)=SQRT(Y*Y+(ALTUD-HEIGHT)**2)
      ANG=ATAN(DERIV)
      RADCRV=((1.+DERIV*DERIV)**1.5)/DERIV2
      CRØØT(K)=SQRT(ABS(RADCRV/((RADCRV-RANGE(K))*RANGE(K))))
      CRØØT(K)=CRØØT(K)*GAIN(ANG-ANGCTR)
6     CØNTINUE
      IF(IRPTW.EQ.0)GØ TØ 4
      IPLØT(1)=20.*REAL(Z(I))/SCALE+233.5
      IPLØT(2)=20.*REAL(ZD(I))/SCALE1+700.
      IPLØT(4)=20.*TANTHT/SCALE1+700.
      IPLØT(3)=20.*REAL(ZDD(I))/SCALE2+1167.5
      DØ 30 J=1,IRPTW
C REPEAT TØ SPREAD SCALE
30    CALL PLØTIT(IPLØT,LAST,I4,IØNE)
4     TMDIF=TANDIF
      NUM=K
C REØRDER IN INCREASING RANGE
10    DØ 7 K=2,NUM
      IF(RANGE(K-1).GT.RANGE(K)) GØ TØ 8
7     CØNTINUE
      GØ TØ 9
8     X=RANGE(K)
      RANGE(K)=RANGE(K-1)
      RANGE(K-1)=X
      X=CRØØT(K)
      CRØØT(K)=CRØØT(K-1)
      CRØØT(K-1)=X
      GØ TØ 10
9     CØNTINUE
C GENERATE ØUTPUTS NØW
      TIME=(RANGE(1)-RNGINC)/C2
      Y=0
      LAST(1)=0
      IPLØT(1)=0
      WRITE(7,21)
21    FØRMAT(1H0)
11    TIME=TIME+DT
      RG=C2*TIME
C RG=RANGE FØR PULSE CENTER
C RGMAX,RGMIN=START,END
      RGMAX=RG+RNGINC
      RGMIN=RG-RNGINC
C FIND REFLECTØR LIMITS,MINY,MAXY
C I.E. THØSE IN THE RANGE BIN
      DØ 12 I=1,NUM
      IF(RANGE(I).GE.RGMIN) GØ TØ 13
12    CØNTINUE
```

```
      ICØUNT=ICØUNT+1
      IF(ICØUNT LT IELKS) GØ TØ 40
      STØP
13    MINV=I
      DØ 14 I=MINV,NUM
      IF(RANGE(I).GT.RGMA<) GØ TØ .5
14    CØNTINUE
      I=NUM+1
15    MAXV=I-1
      ZZ=CMPLX(0.,0.)
C GENERATE ØUTPUTS
C & PLØT THEM
      IF(MAXV LT.MINV)GØ TØ 20
      DØ 16 I=MINV,MAXV
      ANG=WO*(RANGE(I)-RANGE(MINV))/C2
      W=CMPLX(CØS(ANG),SIN(ANG))
      ZZ=ZZ+CRØØT(I)*PULSE(RANGE(I)-RG)*W
16    CØNTINUE
C FILTER
20    X=RHØ*Y+RHØ1*(CABS(ZZ)**2)
      Y=X
      IF(IRPTP.EQ.0)GØ TØ 11
      IPLØT(1)=140C *X*ALTUD
      DØ 24 I=1,IFFTF
24    CALL PLØTIT(IPLØT,LAST,IØNE,IØNE)
18    FØRMAT(1H  4E18 6,4I7)
      GØ TØ 11
      END
.U       EXT  FUNC
IPAN     INT4 VAR
Z        CMPX VAR
ZD       CMPX VAR
ZDD      CMPX VAR
CMPLX    EXT  FUNC
CONJG    EXT  FUNC
W        CMPX VAR
ZZ       CMPX VAR
TEL      REAL VAR
IGFD     INT2 VAR
IPLØT    INT2 VAR
LAST     INT2 VAR
RANGE    REAL VAR
CRØØT    REAL VAR
TAN      STATE FN
.Q       EXT  FUNC
F        EXT  FUNC
X        FORM PAR
SIN      EXT  FUNC
COS      EXT  FUNC
SPEC     STATE FN
'        FØRM PAR
AZ       REAL VAR
```

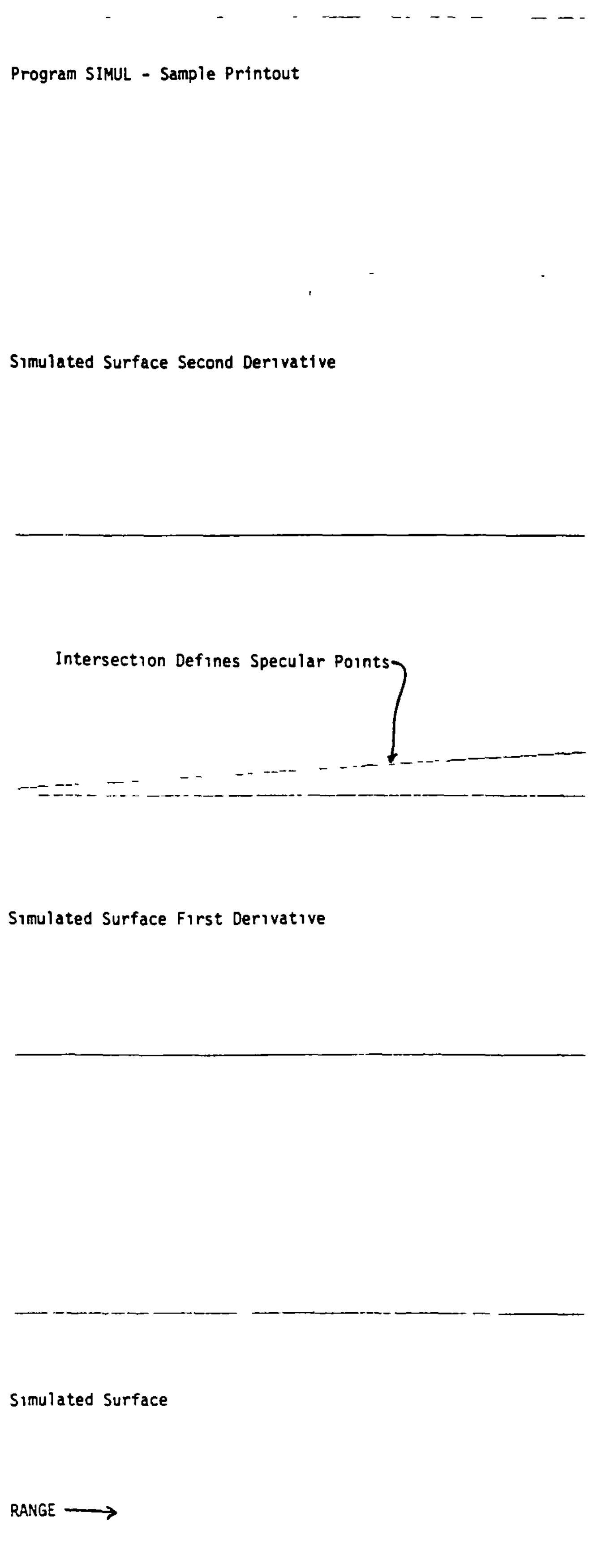
```
EXF      EXT FUNC
AMIN1    EXT FUNC
BG2U4    REAL VAR
.R       EXT FUNC
PULSE    STATE FN
X2       FØRM PAR
SIGPLS   REAL VAR
GAIN     STATE FN
X0       FØRM PAR
ANGVAL   REAL VAR
AMAX1    EXT FUNC
ABS      EXT FUNC
SIGTIM   REAL VAR
NFFT     INT2 VAR
LU       INT2 VAR
P2       REAL VAR
C        REAL VAR
DT       REAL VAR
PULSTM   REAL VAR
F0       REAL VAR
I3       INT2 VAR
I4       INT2 VAR
DTØR     REAL VAR
IØNE     INT2 VAR
IEIG     INT2 VAR
IGDFT3   INT2 VAR
W0       REAL VAR
NFFT2    INT2 VAR
NFFT21   INT2 VAR
25       LABEL
.H       EXT FUNC
26       LABEL
RMESIN   REAL VAR
RMSFHN   REAL VAR
PERIØD   REAL VAR
27       LABEL
FOTS     REAL VAR
31       LABEL
32       LABEL
IPFTW    INT2 VAR
IRFTP    INT2 VAR
41       LABEL
IELKS    INT2 VAR
RHØ      REAL VAR
RHØ1     REAL VAR
SQRT     EXT FUNC
ALTUD    REAL VAR
C2       REAL VAR
RNGINC   REAL VAR
ANGLIN   REAL VAR
ANGCTR   REAL VAR
BMWITH   REAL VAR
```

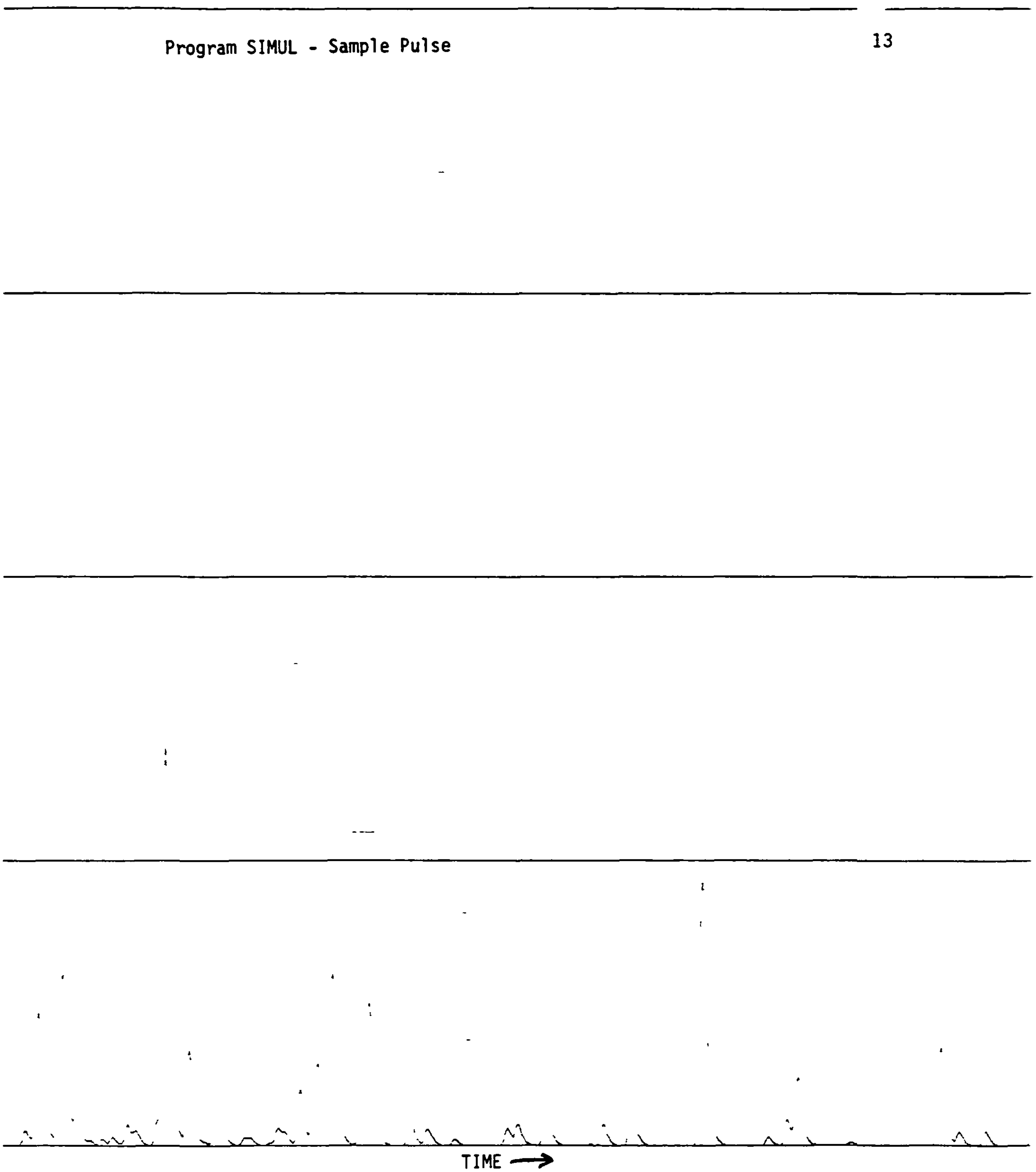
```
ANG3DB   REAL VAR
ANGMAX   REAL VAR
ANGMIN   REAL VAR
Y0       REAL VAR
GDINC    REAL VAR
FLØAT    EXT FUNC
FLØAT2   EXT FUNC
5        LABEL
I        INT2 VAR
.Y       EXT FUNC
Y3DB     REAL VAR
DY       REAL VAR
DY2      REAL VAR
Y02      REAL VAR
DF       REAL VAR
RTDF     REAL VAR
DW       REAL VAR
FØUR1T   EXT FUNC
ICØUNT   INT2 VAR
40       LABEL
$F       EXT FUNC
1        LABEL
X        REAL VAR
RANDG    EXT FUNC
$M       EXT FUNC
IFREQ    INT2 VAR
$L       EXT FUNC
$A       EXT FUNC
SCALE    REAL VAR
SCALE1   REAL VAR
SCALE2   REAL VAR
2        LABEL
XY       REAL VAR
$K       EXT FUNC
3        LABEL
IC       INT2 VAR
SETGFD   EXT FUNC
TMDIF    REAL VAR
REAL     EXT FUNC
K        INT2 VAR
4        LABEL
Y        REAL VAR
TANTHT   REAL VAR
TANHLF   REAL VAR
TANDIF   REAL VAR
6        LABEL
DLT      REAL VAR
DLT1     REAL VAR
HEIGHT   REAL VAR
DERIV    REAL VAR
DERIV2   REAL VAR
ANG      REAL VAR
```

ATHN	EXT FUNC
RADCRV	REAL VAR
.A	EXT FUNC
30	LABEL
J	INT2 VAR
PLØTIT	EXT FUNC
NUM1	INT2 VAR
10	LABEL
7	LABEL
8	LABEL
9	LABEL
TIME	REAL VAR
21	LABEL
11	LABEL
RG	REAL VAR
RGMAX	REAL VAR
RGMIN	REAL VAR
12	LABEL
13	LABEL
.S	EXT FUNC
MINV	INT2 VAR
14	LABEL
15	LABEL
MAXV	INT2 VAR
20	LABEL
16	LABEL
CHES	EXT FUNC
24	LABEL
18	LABEL
.W	EXT FUNC

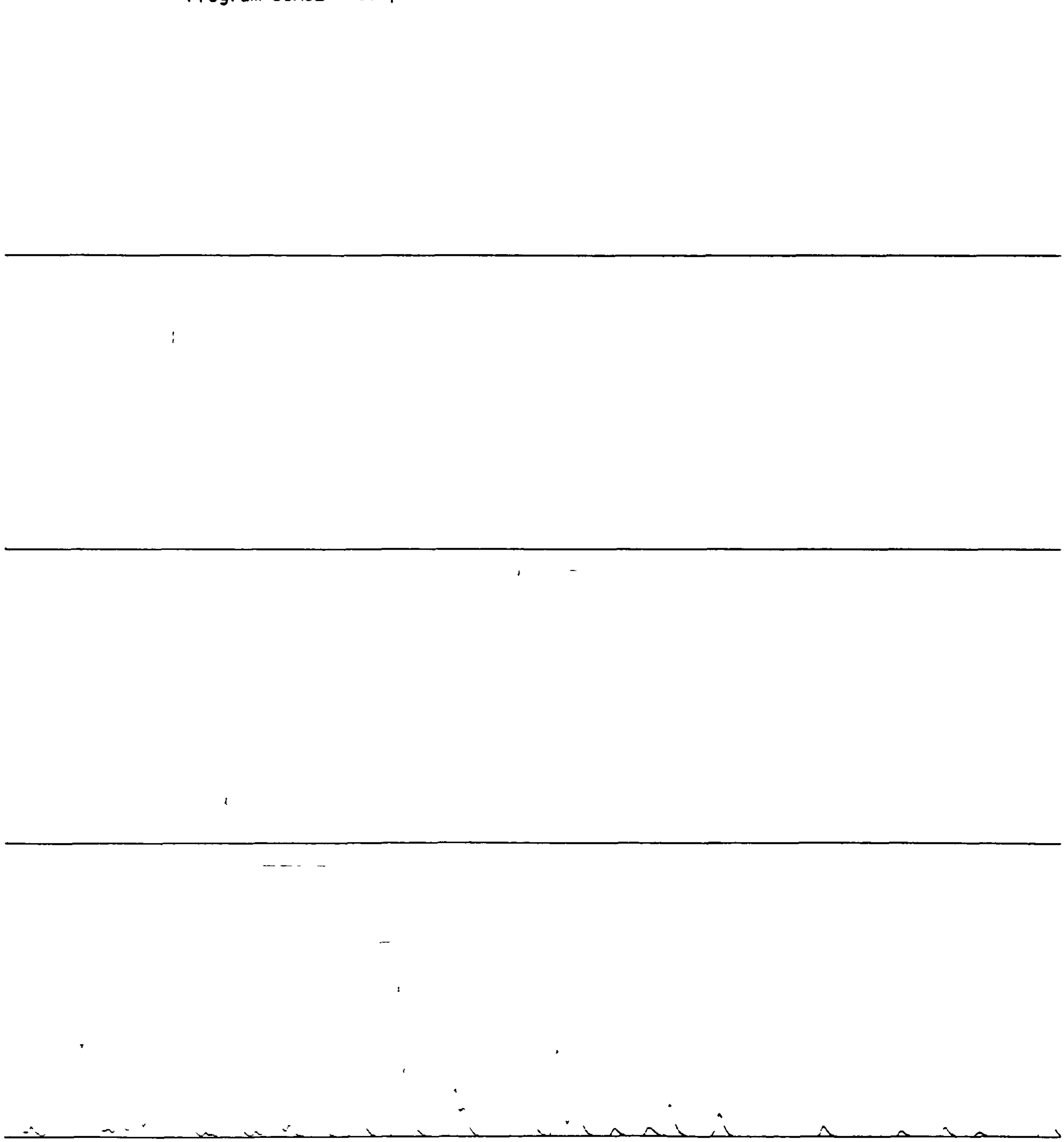
0000 ERRØRS

Program SIMUL - Sample Printout

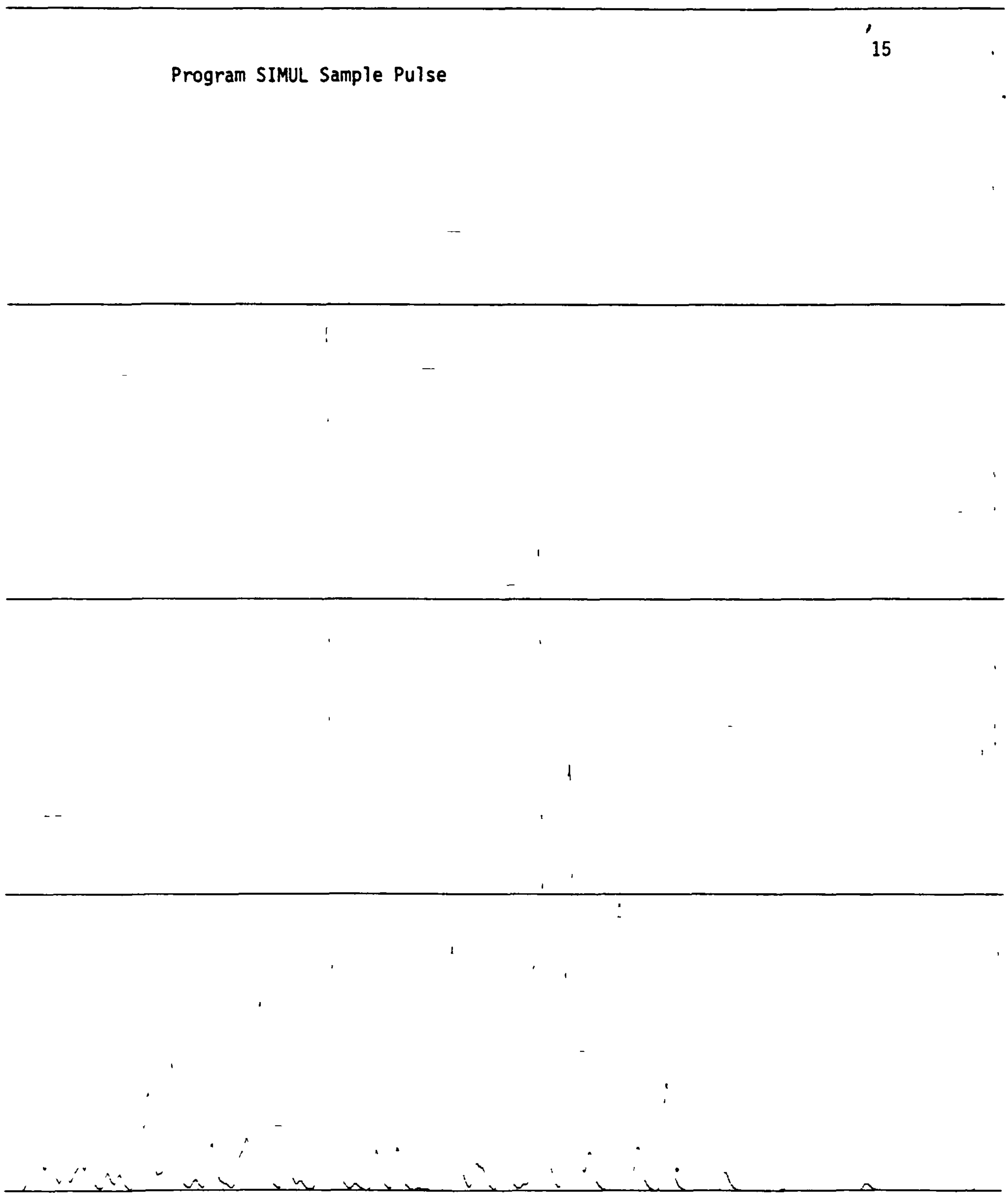



TIME →

Program SIMUL - Sample Pulse



Program SIMUL Sample Pulse



Appendix B

SUBROUTINES

Program Listings

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Program PLOTIT

```
PROGRAM TO PLOT DATA                                    PAGE   1
PROG. PLOTIT

                    SCRAT
      * ENTRIES AT SETGRD AND PLOTIT
      * FORTRAN CALLABLE. SETGRD MUST BE CALLED
      * FIRST TO SET UP GRID.
      *         CALL SETGRD (LU,IGRID,NGRID,NSPACE)
      * WHERE
      * LU=PLOTTER LU
      * IGRID IS AN ARRAY OF Y GRID LOCATIONS
      * NGRID=THE NUMBER OF THESE POINTS
      * NSPACE=NUMBER OF POINTS BETWEEN GRID
      * LINES IN THE X-DIRECTION
      * TO PLOT CALL
      *         CALL PLOTIT(IDATA,LASTDATA,NPOINTS,NCONNECT)
      * WHERE IDATA IS A DATA ARRAY
      * NCONNECT=0 TO PLOT POINTS ONLY
      * NE TO CONNECT THE PRESENT POINT WITH
      * THE LAST
      * NPOINTS=NUMBER OF POINTS IN DATA ARRAY
      * LASTDATA IS AN ARRAY TO HOLD THE LAST
      * POINT
      *
                    ENTRY  SETGRD,PLOTIT
      *
      SETGRD        STM                                 SAVE REGS
                                                        GET NUMBER OF ARGS
                                                        SAVE
                                                        NO OF ARGS
                                                        ERROR EXIT
                                                        SET UP GRID ARRAY
      P1                                                ZERO THIS ONE
                                                        ALL LINES FOR VERTICAL LINE
                                                        GET ARGS
                                                        SET LU
                                                        GET NUMBER OF GRID LINES
      P2                                                QUIT WHEN
                    BNP                     4EC0 2000   GET NEXT POINT
                                                        LIMITS
                                                        SET=0
      F4                                                TOO LARGE
                                                        LIMIT UPPER
      FE                                                SET
      FC                                                SET X SPACES
                                                        FORCE AXIS AT BEGINNING
                                                        EXIT
      *
      * PLOTTING ENTRY POINT
      PLOTIT        STM       10 FE3
                    LH                                  GET ARGS
                    NHI                                 FIX UP R 5
                    ST
                    SIS                                 NUMBER ARGS
                    BNE                                 IF NON ZERO EXIT ON ERROR
                    LH                                  GET ARG LIST
                    LH                                  CONNECT VALUE
                    LH                                  NUMBER OF POINTS
                    SLLS                                2 FOR ENTRY
                    L           COUNT                   X GRID LINE
                    AIS
                    ST
                    CL
                    BNE                                 RESET COUNT
                    ST
      F6            SIS                                 WRITE VERTICAL LINE
                    BM                                  NOW MOVE TO LAST
                    LH                                  UNTIL DONE
```

PROGRAM TO PLOT DATA ... PAGE 2

```
          STH    1,0(13,14)
          B      P16
P14       LHI    11,LINESIZE-3     SET UP OUTPUT BUFF
P15       L      10,GRID(11)       WITH GRID LINES
          ST     10,OUT(11)
          SIS    11,4              DECR COUNT
          BNMS   P15
P6        SIS    14,2              NOW DO DATA POINTS
          BM     P7                UNTIL DONE
          LH     11,0(12,14)       GET POINT
          BNMS   P8                CK LIMITS
          LIS    11,0              LIMIT TO 0 TO
P8        CLHI   11,MAXPTS
          BLS    P9
          LHI    11,MAXPTS-1       MAXPTS-1
P9        LR     15,5              CK FOR CONNECT
       X  BPS    CONNECT           NE MEANS CONNECT
          BMS    BAR               IF - FORCE 0 CONNECT(BAR TYPE)
          STH    11,0(13,14)       ELSE FORCE NO CONNECT BY STORE
CONNECT   LH     10,0(13,14)       GET LAST POINT
          BNMS   P10               CK LIMITS
BAR X     LIS    10,0
P10       CLHI   10,MAXPTS
          BLS    P11
          LHI    10,MAXPTS-1
P11       STH    11,0(13,14)       STORE NEW POINT IN LIST
          CLR    11,10             FORCE (10) LT (11)
          BNLS   P12
          LR     15,10             MOVE
          LR     10,11
          LR     11,15
P12       SBT    10,OUT            SET BITS
          AIS    10,1              UNTIL
          CLR    11,10             (10)=(11)
          BNLS   P12
          B      P6                GET NEXT
P7        SVC    1,WRITE           WRITE IT OUT
P13       LM     10,REG            EXIT
          B      0(15)
*
LINESIZE  EQU    75                BYTES
MAXPTS    EQU    1400              BITS
          ALIG   4
WRITE     DCX    2000
          DAC    OUT,OUT+LINESIZE-1,0,0
WRGRID    DCX    2100
          DAC    GRID,GRID+LINESIZE-1,0,0
REG       DS     20
P1S       DS     4
NXSPACE   DS     4
XCOUNT    DS     4
GRID      DS     LINESIZE+1
CONT      DS     LINESIZE+1
OUT       DS     LINESIZE+1
          END
```

ORIGINAL PAGE IS
OF POOR QUALITY

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR


```
[redacted]
     FROGPAM TO FLOT DATA [illegible]                    PAGE   3

 NØ ERRØRS     3 SCUEB PASSE[illegible]

 CALRO3

    Ø          0000 0[illegible]AI          [illegible]8
   ABSTØP      0000 0030
   ADC         0000 0004
   BHP         0000 01[illegible]I          105
   CONNECT     0000 0[illegible]34I         108
   ERR         00[illegible] 0[illegible]14I         76
   G[illegible]UT        0000 02[illegible]I          4[illegible]   61    55
   [illegible]FID        0000 0[illegible]          44   136   1[illegible]
   INPTØP      0000 0[illegible]
   L[illegible]        [illegible]
   LINESIZE    000[illegible] 0[illegible]          [illegible]2    94   1[illegible]   [illegible]   14[illegible]   144   145
   MH PTS      0000 0F78           58    60   104   10[illegible]   114   116
   N SPACE     0000 0[illegible]94I          64    84
   ØKS         0000 0020I          36
   ØUT         0000 0[illegible]I          96   123   136   136
   P1          0000 0028I          46
   P10         0000 010EI          112
   P11         0000 01[illegible]I          115
   P12         0000 012[illegible]I          119   126
   P1[illegible]         0000 01[illegible]AI          67    90
   P14         0000 00[illegible]          85
   P15         0000 00D2I          98
   [illegible]
   [illegible]                                  62
   [illegible]                                  54
   [illegible]                                  5[illegible]
   [illegible]                                  59
   [illegible]                                  127
   [illegible]        0000 0[illegible]3[illegible]I          100
   [illegible]                                  102
   [illegible]                                  105
   [illegible]
   [illegible]
   [illegible]                                  [illegible]4    74
   [illegible]                                  [illegible]1    [illegible]   12[illegible]
   [illegible]
   [illegible]                                  50    8[illegible]
   [illegible]                                  4[illegible]   12[illegible]
   [illegible]UNT                               [illegible]    [illegible]    [illegible]    [illegible]
```

Program RANDG


```
$HSSM1
RANDG PROG GAUSSIAN RANDOM NUMBER GENERATOR LDD 3/76
 EFLST
$FORT
      FUNCTION RANDG(IY)
C IY IS AN INITIALISING VALUE,ODD,LT 9 DIGITS,
C INTEGER*4  NORMAL NUMBERS GENERATED BY SUMMING
C 12 UNIFORMS IN THE USUAL FASHION. RANDU ROUTINE
C IS USED (NOT CALLED).
C
      X=0.
      DO 1 I=1,12
      IY=IY*65539
      IF(IY.GT.0)GO TO 2
      IY=IY+2147483647+1
2     X=X+FLOAT(IY
1     CONTINUE
      RANDG=X/2147483647.- 6.
      RETURN
      END
RANDG    FUNC/SUB
RANDG    FUNC VAR
.O       EXT FUNC
 P       EXT FUNC
IY       FORM PAR
X        REAL VAR
1        LABEL
I        INT4 VAR
2        LABEL
FLOAT    EXT FUNC

0000 ERRORS
```

Program RANDU

```
$ASSM
RANDU PRØG RANDØM NUMBER GENERATØR LDD 3/76
 EFLST
$FØRT
      FUNCTIØN RANDU(IY)
C NOTE THAT IY IS INTEGER*4
C RESULT IS UNIFØRM [0,1] BY IBM SSP
C IY SHØULD BE INITIALLY ØDD WITH LT 9 DIGITS
      IY=IY*65539
      IF(IY)5,6,6
5     IY=IY+2147483647+1
6     RAND=IY
      RANDU=RAND*.4656613E-9
      RETURN
      END
RANDU    FUNC/SUB
RANDU    FUNC VAR
 Q       EXT FUNC
 P       EXT FUNC
IY       FURM PAF
5        LABEL
6        LABEL
RAND     REAL VAR
.W       EXT FUNC

0000 EPROPS
```