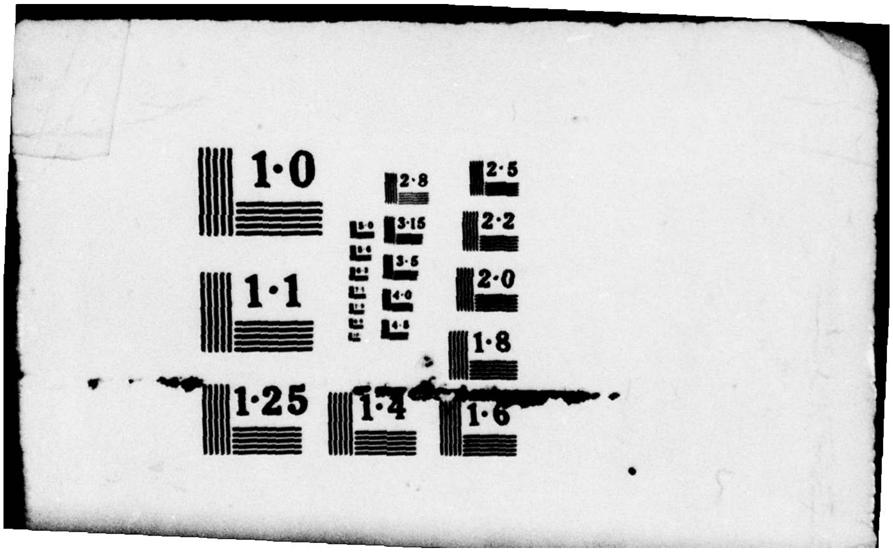
AD-A073 051   UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES   DEPT O--ETC   F/G 9/2
SOFTWARE FOR NONLINEAR FILTERING.(U)
JUN 79   R S BUCY, F GHOVANLOU   F44620-76-C-0085

UNCLASSIFIED   USCAE-53-4514-1787   AFOSR-TR-79-0923   NL

1 OF 2
AD
A07305


1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
1·1
4·5
1·8
1·25
1·4
1·6

UNIVERSITY OF SOUTHERN CALIFORNIA

SCHOOL OF ENGINEERING

SOFTWARE FOR NONLINEAR FILTERING

R.S. Bucy, F. Ghovanlou
A.J. Mallinckrodt, K.D. Senne

University of Southern California
Los Angeles, California 90007

DEPARTMENT OF AEROSPACE ENGINEERING

"SOFTWARE FOR NONLINEAR FILTERING"[1]

R.S. Bucy[2], F. Ghovanlou[2]
A.J. Mallinckrodt[3], K.D. Senne[4]

[1] This research was supported in part by the United States Air Force Office of Scientific Research under Contract F44620-76-C-0085, and Grant AFOSR 76-3100.

[2] University of Southern California, Los Angeles, California.

[3] Communications Research Laboratory, Santa Ana, California.

[4] MIT, Lincoln Laboratory, Lexington, Massachusetts.

Table of Contents

Introduction

As part of a continuing search for the ideal architecture for performing the computations required to realize a non-linear filter, we have developed software for various machines over the past ten years. A description of the latest software is given in [1] , while [2] , [3] , and [4] are useful for background information on the non-linear filtering problem as well as comments about software efficiencies relevant to various machines.

We started our studies over 10 years ago using the CDC 6600 at the Aerospace Corporation and Kirkland AFB, and continuing at Eglin AFB, see [4] . At the Institute for Advanced Computation, we gained access to the Illiac IV and at ICASE, Nasa Langley , the Star 100, see [2] . Access to the Cray was obtained through Cray Research and later at NCAR. Experiments on the AP120B array processor were possible because of the acquisition of one here at USC used in conjunction with a PDP 11-55.

The purpose of this report is to document the current software, for all these machines. In particular, we have found [2] , with the listings of the 6600 and Star Codes, extremely useful in the past, although now these listings are outdated. In particular, the assembly language coding for the AP-120B involved extensive effort over a long time period and should be documented so that others interested in similar problems, can avoid the pain of developing the software from scratch.

I PHASE DEMODULATION

I - 1 CDC 6600 Code

The code shown in the following pages evolved through a number of changes. It was most effected by the coding of the Star given in the next section. The philosophy was; carry the two-dimensional density as a single vector of array columns and break up the computation into a large number of loops each small enough so that at least inner loops fit into the stack. Using the CDC FTN Compiler Opt = 2, level 410 operating system this code achieves .63 megaflops.

A. TWO-DIMENSIONAL CDC-6600 PROGRAM

```
C <BUCY>STAF.FOR;1       4-NOV-76 10:01:44     EDIT BY BUCY
      PROGRAM CYCLIC(INPUT=129,OUTPUT=129,TAPE5=INPUT,TAPE6=OUTPUT)
C         DESCRIPTION OF INPUT PARAMETERS
C
C                 Y1EST,Y2EST - THE EXPECTED VALUE OF INITIAL POSITION
C                 ALP110 - STEADY STATE ERROR VARIANCE IN DECIBELS
C                 DELF - THE RATIO OF DELTA TO FILTER TIME CONSTANT
C                 Q22C - THE CONTINUOUS DRIVING VARIANCE
C                 NUM1,NUM2 - ARE USED IN CYCLIC AND PROBE ONLY AND COUNT T
C                               NUMBER OF PARTITION POINTS IN RECTANGULAR GRI
C                 NO2 - THE TOTAL NUMBER OF POINTS (ESTIMATES) IN EACH SAMP
C
C
C         DESCRIPTION OF DATA SET
C
C                 DATA MUST BE PUNCHED IN THE FOLLOWING ORDER:
C
C                     Y1EST,Y2EST,ALP110,DELF,Q22C,NUM1,NUM2,NO2
C
C                 ALL REAL PARAMETERS (Y1EST THRU Q22C) HAVE A 10 SPACE FIE
C                 ALL INTEGER PARAMETERS (NUM1 THRU NO2) HAVE A 5 SPACE FIE
C                 AND MUST BE RIGHT JUSTIFIED IN THIER RESPECTIVE FIELDS.
C
C
C         COMMENTS
C
C                 THE MAIN FLOW THROUGH THE PROGRAM IS GOVERNED BY KOUNT.
C                 KOUNT COUNTS THE POINTS IN EACH PATH.  A BLOCK IS A SECTI
C                 OF THE PROGRAM THAT HAS NO TRANSFER IN OR OUT EXCEPT
C                 THROUGH COMMON.
C
C
C         ***************************************************************
C
      COMMON XDAT(130,5)  ,XHAT(2)
      COMMON P(2,2),PBAR(2,2),PN(2,2),AN(2),F(2,2),PDUMY(2,2),
     *PDUMY2(2,2),PNF(2,2)
      LOGICAL LOW,UP
      COMMON /RN/ NZZZ(3), XNZ(2)
      COMMON /GN/ DZZZ1, JGAUSS, XZZZ(2)
      COMMON /PROB/ PI2,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,
     1    A11,A22,CONST,DELT,FTC,PIDLT,P110,R11,RX,Q0,Q22
C     ****************************  START BLOCK 1  ***********************
    2 CONTINUE
      JGAUSS=0
      Y1EST=0.0
      Y2EST=0.0
      ALP110=-3.00
      DELF=0.1
      Q22C=0.01
      NUM1=33
      NUM2=127
      NO2=130
      NO3=1
      IF(EOF(5)) 2200,5
      PHASE VARIABLES
      DO 210 I=1,32
      SIGMA(I)=PI*((2.*I-1.)/32.-1.)
      COSY(I)=COS(SIGMA(I))
      SINY(I)=SIN(SIGMA(I))
      S1(I)=COSY(I)/RDEL
  210 S2(I)=SINY(I)/RDEL
      PHASE RATE VARIABLES
      DO 220 I=1,128
  220 PSI(I)=PIDEL*((2.*I-1.)/128.-1.)
      SETUP THE TRANSFER MATRIX
      DO 240 J=1,128
      J1=(J-1)*32
      J2=(J-1)*33
      DO 230 I=1,32
      I1=J1+1+MOD(46-(J-1)/4+I,32)
      I2=J2+I
  230 JNS(I2)=I1
  240 JNS(J2+33)=JNS(J2+1)
      SETUP THE INTERPOLATION VECTOR
      IN(1)=0.875
      IN(2)=0.625
      IN(3)=0.375
      IN(4)=0.125
      IN(5)=IN(1)
      IN(6)=IN(2)
      IN(7)=IN(3)
      IN(8)=IN(4)
      J=MOD(NTERM,4)
      DO 245 I=1,4
  245 DELJ(I)=IN(I)
      DO 250 I=5,125,4
      DELJ(I)=DELJ(I-4)
      DELJ(I+1)=DELJ(I-3)
      DELJ(I+2)=DELJ(I-2)
  250 DELJ(I+3)=DELJ(I-1)
      EVALUATE CONVOLUTION TERMS A(I)
      DO 280 I=1,NTERM
     'TEMP=I/128.
      TEMP=CONST*TEMP*TEMP
      A(I)=0.
      IF (TEMP.GT.-47) A(I)=EXP(TEMP)
  280 CONTINUE
      CONSTRUCT THE A PRIORI DENSITY
      CNORM=1.0/(TWOPI*SQRT(A11*A22))
      CL=-0.5/A22
      SI=-0.5/A11
      DO 290 I=1,32
      CP=SIGMA(I)-Y1EST
      CP=CP*CP*SI
      J1=0
      DO 290 J=1,128
      J2=J1+I
      TEMP=PSI(J)-Y2EST
      JC(J2)=EXP(TEMP*TEMP*CL+CP)*CNORM
  290 J1=J1+32
      RETURN
      J1=NSIZE32
      J2=J1
      DO 60 I=1,NTERM
      J1=J1+32
      J2=J2-32
      TEMP=A(I)
      DO 60 J=1,4096
      K1=J1+J
      K2=J2+J
   60 JN(J)=JN(J)+TEMP*(JNA(K1)+JNA(K2))
C     CUMULATE ROW SUMS
      DO 80 I=1,32
      I1=I
      TEMP2=JN(I1)
      DO 70 J=1,127
      I1=I1+32
   70 TEMP2=TEMP2+JN(I1)
   80 TROW(I)=TEMP2
C     ACCUMULATE ESTIMATES AND NORMALIZATION CONSTANT
      CNORM=TROW(1)*SN1(1)
      SHAT=SINY(1)*CNORM
      CHAT=COSY(1)*CNORM
      DO 85 I=2,32
      TEMP2=TROW(I)*SN1(I)
      SHAT=SHAT+SINY(I)*TEMP2
      CHAT=CHAT+COSY(I)*TEMP2
   85 CNORM=CNORM+TEMP2
      CNORM=1.0/CNORM
      SHAT=SHAT*CNORM
      CHAT=CHAT*CNORM
C     TRANSFER NORMALIZED DENSITY
      DO 90 I=1,32
      I1=I
      TEMP2=SN1(I)*CNORM
      DO 90 J=1,128
      JN(I1)=TEMP2*JN(I1)
   90 I1=I1+32
C     TIMEOUT
      TSLF=SECOND(TT)-T
      RETURN
C     INITIALIZE SAMPLE PATH BY TRANSFERING J0 TO JN
  100 IF (MC.LE.0) GO TO 200
      DO 110 I=1,4096
  110 JN(I)=J0(I)
      RETURN
C     GLOBAL INITIALIZATIONS FOR NONLINEAR FILTER
  200 NSIZE=10
      NTERM=64.0*SQRT(50.*Q22)/PIDEL+0.5
      IF (NTERM.GT.NSIZE) NTERM=NSIZE
      NSIZE32=NSIZE*32
      NTERM32=NTERM*32
      NK2=NSIZE32+1
      NJ1=NK2-NTERM32
      NJ2=NSIZE32+4097-NTERM32
      NK1=NSIZE32+4097
      SUBROUTINE NLF(MC,SAMP,Z1,Z2,SHAT,CHAT,TNLF)
      INTEGER MC,SAMP
      REAL Z1,Z2,SHAT,CHAT,TNLF
      INTEGER I,I1,JC,J1,J2,K1,K2,KL,KH,NJ1,NJ2,NK1,NK2,NTERM,
     1   NTERM32,NSIZE,NSIZE32
      REAL A11,A22,CL,CNORM,CONST,CR,PI,PIDEL,Q22,SI,T,TT,
     1   Y1EST,Y2EST,TEMP,TEMP1,TEMP2
      REAL IN(8)
      REAL TROW(32)
      REAL COSY(32),SINY(32),SN1(32),S1(32),S2(32),SIGMA(32)
      REAL PSI(128),A(10),DELJ(128)
      INTEGER JNS(4224)
      REAL JN(4096),JN1(4096),J0(4096),JNA(4756)
      COMMON /PROB/ TWOPI,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,
     1   A11,A22,CONST,DEL,FTC,PIDEL,P110,RDEL,RX,QC,Q22
      COMMON /NLFC/ NC,NT,NTERM,NTERM33,S1,S2,SIGMA,PSI,A,COSY,
     1   DELJ,J0,JNA,JNS,SINY
      EQUIVALENCE (JN1(1),JNA(321))
      IF (SAMP.LE.0) GO TO 100
      SET CLOCK
      T=SECOND(T)
      EVALUATE SENSOR TERMS
      DO 10 I=1,32
   10 SN1(I)=EXP(Z1*S1(I)+Z2*S2(I))
      FORM THE INTERPOLATED JN AND PUT IN JN1
      J1=0
      J2=0
      DO 30 I=1,128
      TEMP=DELJ(I)
      DO 20 J=1,32
      K1=J1+J
      KL=JNS(K1)
      KH=JNS(K1+1)
      TEMP1=JN(KL)
      K2=J2+J
   20 JN1(K2)=TEMP1+TEMP*(JN(KH)-TEMP1)
      J1=J1+33
   30 J2=J2+32
      EXPAND INTERPOLATED MATRIX ON BOTH SIDES
      J1=NJ1
      J2=NJ2
      K1=NK1
      K2=NK2
      DO 40 I=1,NTERM32
      JNA(J1)=JNA(J2)
      JNA(K1)=JNA(K2)
      J1=J1+1
      J2=J2+1
      K1=K1+1
   40 K2=K2+1
      CONVOLUTION
      DO 50 I=1,4096
      J=I+NSIZE32
   50 JN(I)=JNA(J)
```

```
      SUBROUTINE GAUSS(JS,SD,XM,X)
      DIMENSION  NST(2)
      COMMON /RN/ N1, N2, MC, T1, T2
      COMMON /GN/ TWOPI, J, XR(2)
      IF (J) 10, 10, 20
   10 J=2
      TWOPI=8.*ATAN(1.)
      NST(1)=102943
      NST(2)=135617
      XR(1)=BANF(NST,1)
      GO TO 35
   20 GO TO (30,40), J
   30 J=2
      XR(1)=BANF(NST,0)
   35 XR(2)=BANF(NST,0)
      X1=SQRT(ABS(-2.*ALOG(XR(1))))
      XR(2)=TWOPI*XR(2)
      XR(1)=X1*SIN(XR(2))
      XR(2)=X1*COS(XR(2))
      X=XR(1)*SD+XM
      RETURN
   40 J=1
      X=XR(2)*SD+XM
      RETURN
      END
```

```
      FUNCTION BANF(NS,MODE)
      DIMENSION  NS(2), NC(2)
      COMMON /RN/ N1, N2, MP, T1, T2
      DATA  M1, M2/244734, 158551/
C        MODE=0 TO CONTINUE, OTHERWISE RESTART WITH
C        INTEGER NUMBER NS(1)*2**18+NS(2)
      IF (MODE) 10, 100, 10
   10 N1=NS(1)
      N2=NS(2)
      T1=2.**(-18)
      T2=2.**(-36)
      MP=2**18
  100 DO 200 I=1,2
      GO TO (110,120),I
  110 K=M2*N2
      GO TO 190
  120 K=M1*N2+M2*N1+KD
  190 KD=K/MP
  200 NC(I)=K-KD*MP
      N1=NC(2)
      N2=NC(1)
      XN1=N1
      XN2=N2
      BANF=XN1*T1+XN2*T2
      RETURN
      END
      H=NO2-30
      SUMP=SUMP/H
      SUMC=SUMC/H
       XNSAMP=NSAMP
       XAA=XNSAMP+1.0
      SUMP1=(SUMP+XNSAMP*SUMP1)/XAA
      DSUMP1=ALOG10(SUMP1)*10.
      WRITE(6,1508)
 1508 FORMAT(*0*,5X,*NONLINEAR CYCLIC ESTIMATOR*)
      WRITE(6,1511)SUMP1,DSUMP1
 1511 FORMAT(*0*,*AVERAGE STATISTICAL VARIANCE =*,1PE13.6, 10X,
     * *AVERAGE COMPUTED VARIANCE =*,1PE13.6//)
      SUMP=0.0
      SUMC=0.0
      DO 1601 I=31,NO2
        XD=ABS(XDAT(I,1)-XDAT(I,4))
 1698   CONTINUE
        IF(XD.GT.PI) GO TO 1699
        GO TO 1700
 1699   XD=XD-PI2
        GO TO 1698
 1700   SUMP=(XD)**2+SUMP
        SUMC=XDAT(I,5)+SUMC
 1601   CONTINUE
      SUMP=SUMP/H
      SUMC=SUMC/H
      SUMP2=(SUMP+XNSAMP*SUMP2)/XAA
      DSUMP2=ALOG10(SUMP2)*10.
      WRITE(6,1509)
 1509 FORMAT(*0*,5X,*RE-LINEARIZED K-B FILTER*)
      WRITE(6,1511)SUMP2,DSUMP2
      NSAMP=NSAMP+1
      IF(ISAMP.EQ.NO3)GO TO 2200
      ISAMP=ISAMP+1
      GO TO 11
      ******************************  END BLOCK 3  ************************
 2200 WRITE(6,2201)
 2201 FORMAT (*0*,40X,*NORMAL COMPLETION*)
      STOP
      END
 88 X1FF=X1FF-PI2
    GO TO 84
 89 X1FF=X1FF+PI2
    GO TO 84
 90 CONTINUE
    IF(ABS(CXHAT).GT.XLIM)LIMNL=LIMNL+1
    IF(ABS(X1FF).GT.XLIM)LIMKB=LIMKB+1
C   ***************************  PREDICTOR UPDATE  ********************
    XHAT1=XHAT(1) + DELT*XHAT(2)
    XHAT2=XHAT(2)
    XDAT(KOUNT,4)=XHAT(1)
    XDAT(KOUNT,5)=PNF(1,1)
    X1MOD=X1
184 CONTINUE
    IF(X1MOD.GT.PI) GO TO 188
    IF(X1MOD.LT.-PI) GO TO 189
    GO TO 190
188 X1MOD=X1MOD-PI2
    GO TO 184
189 X1MOD=X1MOD+PI2
    GO TO 184
190 CONTINUE
    IKBSLP=0
    X1F2=ABS(XHAT(1)-X1)
339 IF(X1F2.GT.PI) GO TO 340
    GO TO 341
340 CONTINUE
    X1F2=X1F2-PI2
    IKBSLP = IKBSLP+1
    GO TO 339
341 CONTINUE
    ERRLF=ABS(X1FF-X1MOD)
    ERRNL=ABS(CXHAT-X1MOD)
    IF(ERRLF.GT.PI) ERRLF=ABS(ERRLF -PI2)
    IF(ERRNL.GT.PI) ERRNL=ABS(ERRNL-PI2)
    ERROF=ABS(X1MOD-CXHAT)
    WRITE(6,201)KOUNT,XDAT(KOUNT,1),X1MOD,XDAT(KOUNT,2),Z1,Z2, (XDAT
   *(KOUNT,I),I=3,5)
201 FORMAT(*0*,I3,1X,1P3E14.6,4X,1P2E14.6,4X,1P3E14.6 /)
    IF(KOUNT.EQ.NO2) GO TO 505
    KOUNT=KOUNT + 1
    GO TO 450
505 CONTINUE
    SUMP=0.0
    SUMC=0.0
    DO 1501 I=31,NO2
      XD=ABS(XDAT(I,1)-XDAT(I,2))
1498  CONTINUE
      IF(XD.GT.PI) GO TO 1499
      GO TO 1500
1499  XD=XD-PI2
      GO TO 1498
1500  SUMP=(XD)**2+SUMP
      SUMC=XDAT(I,3)+SUMC
1501  CONTINUE
```

```
      DEV3= SQRT(R11)
      CALL GAUSS(JSEED,DEV1,Y1EST,X1)
      KOUNT=1
      XDAT(KOUNT,1)=X1
      CALL GAUSS(JSEED,DEV2,Y2EST,X2)
      CALL GAUSS(JSEED,DEV3,COS(X1),Z1)
      CALL GAUSS(JSEED,DEV3,SIN(X1),Z2)
      DEVQ2= SQRT(Q22)
      R=R11
      WRITE(6,1509)
 205  FORMAT(*0*,8X,*POSIT.*,5X,*POSIT. MOD 2 PI*,2X,*EST. POSIT.*,9X,
     **Z1 AND Z2*,19X,*CYCLIC LOSS*,5X,* K-B EST. AND P11*)
      GO TO 470
C     *****************************  END BLOCK 1  ************************
C     ****************************  START BLOCK 2  **********************
  450 CONTINUE
      X1=X1 + X2*DELT
      XDAT(KOUNT,1)=X1
      CALL GAUSS(JSEED,DEVQ2,X2,X2)
      CALL GAUSS(JSEED,DEV3,COS(X1),Z1)
      CALL GAUSS(JSEED,DEV3,SIN(X1),Z2)
C     ************************  RICCATI EQUATION UPDATE  ****************
      PDUMY(1,1)=(R*(PN(1,1)+2.0*PN(1,2)*DELT)-PN(1,2)**2*DELSQ)*DEN
     *  + PN(2,2)*DELSQ
      PDUMY(1,2)=PN(1,2)*(R-PN(1,2)*DELT)*DEN + PN(2,2)*DELT
      PDUMY(2,2)=-PN(1,2)**2*DEN + PN(2,2) + Q(2,2)
      PN(1,1)=PDUMY(1,1)
      PN(1,2)=PDUMY(1,2)
      PN(2,2)=PDUMY(2,2)
      PN(2,1)=PN(1,2)
      DEN =1.0/(PN(1,1) + R)
      *****************************  END BLOCK 2  ************************
      ****************************  START BLOCK 3  **********************
  470 CONTINUE
      CALL NLF(1,1,Z1,Z2,SHAT,CHAT,TNLF)
      WRITE (6,5697) TNLF
 5637 FORMAT(F10.5)
      CXHAT = ATAN2(SHAT,CHAT)
  367 PLOSS=2.0*(1.0-SQRT(SHAT**2+CHAT**2))
      XDAT(KOUNT,2)=CXHAT
      XDAT(KOUNT,3)=PLOSS
      PNF(1,1)=PN(1,1)*R*DEN
      PNF(1,2)=PN(1,2)*R*DEN
      PNF(2,1)=PNF(1,2)
      PNF(2,2)=PN(2,2) - PN(1,2)**2*DEN
      *****************************  FILTER UPDATE  *********************
      SINF1=SIN(XHAT1)
      COSF1=COS(XHAT1)
      XHAT(1)=XHAT1+DEN*(-PN(1,1)*SINF1*Z1+PN(1,1)*COSF1*Z2)
      XHAT(2)=XHAT2+DEN*(-PN(1,2)*SINF1*Z1+PN(1,2)*COSF1*Z2)
      X1FF=XHAT(1)
   34 CONTINUE
      IF(X1FF.GT.PI) GO TO 88
      IF(X1FF.LT.-PI) GO TO 89
      GO TO 99
```

```
5    CONTINUE
     WRITE(6,651) Y1EST,Y2EST,ALP110,DELF,Q22C,NUM1,NUM2,NO2
651  FORMAT(* *,*  CYCLIC INPUT*,4X,5F10.5,3I5)
     P110=10.**(ALP110/10.)
     QQ=Q22C**(.25)
     RX=(P110/(SQRT(2.0)*QQ))**(4.0/3.0)
     FTC=SQRT(2.0)* RX**(.25) /QQ
     DELT=DELF*FTC
     Q22=Q22C*DELT
     R11=RX/DELT
     P220=P110*SQRT(Q22C/RX)
      ISAMP=1
      NSAMP=0
      SUMP1=0.0
      SUMP2=0.0
      CONTINUE
     A11=10.**((ALP110+1.4)/10.)
     A22=     P220
     KOUNT=1
     DELSQ=DELT**2
     PI=3.1415926536
     PI2=2.0*PI
     PIDLT=PI/DELT
     CONST=-2.0*PIDLT*PIDLT/Q22
     PINV=1.0/PI
     PI2DLT=2.0*PIDLT
     U1=NUM1
     U2=NUM2
     XLIM=.75*PI
     LIMNL=0
     LIMK3=0
     Q(1,1)=0.0
     Q(2,2)=Q22
     A=DELT*PINV*SQRT(10.0*Q22)
     IA=A+0.5
     IY2=U2/PI2DLT*SQRT(50.0*Q22) + .5
     CALL NLF(0,C,Z1,Z2,SHAT,CHAT,TNLF)
 11  CALL NLF(1,C,Z1,Z2,SHAT,CHAT,TNLF)
     XHAT(1)=Y1EST
     XHAT(2)=Y2EST
     XHAT1=Y1EST
     XHAT2=Y2EST
     PN(1,1)=A11
     PN(2,2)=A22
     PN(1,2)=0.
     PN(2,1)=0.
     R=R11
     DEV1= SQRT(A11)
     DEN=1.0/(PN(1,1)+R)
     F(1,1)=1.0
     F(1,2)=DELT
     F(2,1)=0.
     F(2,2)=1.0
     DEV2= SQRT(A22)
```

I - 2 Star 100 Code

This code was developed by keeping in mind that Star is efficient on long vectors and has a large memory bandwidth, consequently the density was carried as a long vector with extra elements carried in the vector to eliminate the need for modular arithmetic. All operations were viewed as column oriented and assembly listing with loop timing were used to iteratively improve the code. Star Fortran is standard Fortran with added vector instructions such as VGATHER, VSUM, etc. The CDC Star Fortran manual will be helpful in understanding the resulting code. Writing this code and tailoring it to the Star strengths provided much insight into our problem and produced significant improvements on code for the other machines. In particular it is strange that coding for the Illiac had little fallout for other machine coding. The code achieved 16 megaflops.

```
FORTRAN P1.2 CYCLE 115P2     O=B              SOURCE LISTING          13.39 HRS. 31MAY77
             PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      C      SAMPLE PATH VARIABLES
             REAL SX1(130),SCHAT(130),SSHAT(130),SX1HATNL(130),SERRNL(130),
            1 TNLF(130),SPLOSSNL(130),SX1HATPL(130),SERRPL(130),SP11PL(130)
      C      CUMULATIVE  SAMPLE PATH VARIABLES
             REAL CERRNL(130),CESQNL(130),CEVARNL(130),CDBNL(130),CCSNL(130)
             REAL CERRPL(130),CESQPL(130),CEVARPL(130),CDBPL(130),CCSPL(130)
      C      MONTE CARLO SUMMARY STATISTICS
             REAL XERRNL,XESQNL,XEVARNL,XDBNL,XCSNL,
            1       XERRPL,XESQPL,XEVARPL,XDBPL,XCSPL
      C      SINGLE SAMPLE VARIABLES
             REAL CHAT,SHAT,X1HAT,P11,X1,Z1,Z2
      C      CONSTANTS
             REAL CS(130),DBEPS,EPS(130),TWOPI,ZERO(130),ONE(130),PI2(130)
      C      WORKING VARIABLES
             INTEGER I,J,K,L
             REAL T,TEMP(130)
             LOGICAL PATH,CUMPATH
             BIT BT(130)
      C      PROBLEM SETUP VARIABLES
             REAL ALP110,DELF,Q22C,Y1EST,Y2EST
             INTEGER NMC,NSAMP,MD,ND
      C      DERIVED PROBLEM CONSTANTS
             REAL A11,A22,CONST,DEL,FTC,PI,PIDEL,P110,RDEL,RX,QQ,Q22
      C      PROBLEM COMMON
             COMMON /PROB/ TWOPI,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,A11,A22,
            1   CONST,DEL,FTC,PIDEL,P110,RDEL,RX,QQ,Q22,MD,ND
      C
      C
             WRITE(6,991)
       991      FORMAT('  FILT2NN, VERSION 4-22')
      C      SET PRINTOUT CONTROL
             PATH=.TRUE.
             CUMPATH=.TRUE.
      C      READ INPUT PARAMETERS
          10 READ (5,5000,END=500) Y1EST,Y2EST,ALP110,DELF,Q22C,NMC,NSAMP,MD
        5000 FORMAT(5E10.4,3I5)
      C      COMPUTE THE CONSTANTS
             MD=(MD/2)*2
             IF (MD.LT.20) MD=32
             IF (MD.GT.64) MD=32
             ND=4*MD
             PI=4.0*ATAN(1.0)
             TWOPI=2.0*PI
             P110=10.**(ALP110/10.)
```

```
FORTRAN R1.2 CYCLE 115P2   O=B          SOURCE LISTING          13.39 HRS. 31MAY7
      QQ=Q22C**(.25)
      RX=(P110/(SQRT(2.0)*QQ))**(4.0/3.0)
      FTC=SQRT(2.0)*RX**(.25)/QQ
      DEL=DELF*FTC
      Q22=Q22C*DEL
      PIDEL=PI/DEL
      RDEL=RX/DEL
      A11=10.0**((ALP110+1.4)/10.)
      A22=2.0*P110/(FTC*FTC)
      CONST=-2.0*PIDEL*PIDEL/Q22
      CS(1;130)=0.75*PI
      PI2(1;130)=TWOPI
      ZERO(1;130)=0.
      EPS(1;130)=1.E-50
      DBEPS=ALOG10(EPS(1))
      ONE(1;130)=1.
C     INITIALIZE CUMULATIVE SAMPLE PATH VARIABLES
      CERRNL(1;130)=0.
      CESQNL(1;130)=0.
      CCSNL(1;130)=0.
      CERRPL(1;130)=0.
      CESQPL(1;130)=0.
      CCSPL(1;130)=0.
C     PRINTOUT PROBLEM PARAMETERS
      WRITE (6,6000) NMC,NSAMP,ALP110,DELF,Q22C,Y1EST,Y2EST,
     1   P110,QQ,RX,FTC,DEL,Q22,PIDEL,RDEL,A11,A22,CONST,CS(1)
 6000 FORMAT(1H1,31X,18HPROBLEM PARAMETERS/
     1   1H0,11X,9HPARAMETER,6X,5HVALUE,11X,9HPARAMETER,
     2   6X,5HVALUE/1H0,14X,3HNMC,9X,I4,14X,5HNSAMP,8X,I4/
     3   13X,6HALP110,3X,E15.8,8X,4HDELF,4X,E15.8/
     4   14X,4HQ22C,4X,E15.8,8X,5HY1EST,3X,E15.8/
     5   14X,5HY2EST,3X,E15.8,8X,4HP110,4X,E15.8/
     6   15X,2HQQ,5X,E15.8,9X,2HRX,5X,E15.8/
     7   15X,3HFTC,4X,E15.8,9X,3HDEL,4X,E15.8/
     8   15X,3HQ22,4X,E15.8,8X,5HPIDEL,3X,E15.8/
     9   14X,4HRDEL,4X,E15.8,9X,3HA11,4X,E15.8/
     A   15X,3HA22,4X,E15.8,8X,5HCONST,3X,E15.8/
     B   15X,2HCS,5X,E15.8)
C     BEGIN THE MONTE CARLO PROCESS
C     INITIALIZATION OF THE SUBROUTINES
      CALL STATE(0,0,X1,Z1,Z2)
      CALL NLF(0,0,0.,0.,SHAT,CHAT,T)
C     MONTE CARLO LOOP
      DO 200 K=1,NMC
C     INITIALIZATION OF SAMPLE PATH VARIABLES
```

```
        CALL STATE(K,0,X1,Z1,Z2)
        CALL NLF(K,0,Z1,Z2,SHAT,CHAT,T)
        CALL PLL(K,0,Z1,Z2,X1HAT,P11)
    C   SAMPLE PATH LOOP
        DO 100 J=1,NSAMP
        CALL STATE(K,J,X1,Z1,Z2)
        CALL NLF(K,J,Z1,Z2,SHAT,CHAT,T)
        CALL PLL(K,J,Z1,Z2,X1HAT,P11)
    C   STORE THE SAMPLE VARIABLES
        SX1(J)=X1
        SSHAT(J)=SHAT
        SCHAT(J)=CHAT
        TNLF(J)=T
        SX1HATPL(J)=X1HAT
    100 SP11PL(J)=P11
    C   VECTOR ACCUMULATE THE SAMPLE PATH AVERAGES
        SX1HATNL(1;130)=VATAN2(SSHAT(1;130),SCHAT(1;130);
       1   SX1HATNL(1;130))
        SERRNL(1;130)=SX1(1;130)-SX1HATNL(1;130)
        CALL MOD2PI(SERRNL)
        SPLOSSNL(1;130)=SSHAT(1;130)*SSHAT(1;130)+SCHAT(1;130)*
       1   SCHAT(1;130)
        SPLOSSNL(1;130)=2.0*(1.0-VSQRT(SPLOSSNL(1;130);SPLOSSNL(1;130)))
        CERRNL(1;130)=VAVG(CERRNL(1;130),SERRNL(1;130),K;CERRNL(1;130))
        TEMP(1;130)=SERRNL(1;130)*SERRNL(1;130)
        CESQNL(1;130)=VAVG(CESQNL(1;130),TEMP(1;130),K;CESQNL(1;130))
        IF (K.LE.1) GO TO 110
        CEVARNL(1;130)=CESQNL(1;130)-CERRNL(1;130)*CERRNL(1;130)
        BT(1;130)=(CEVARNL(1;130).LE.EPS(1;130))
        CEVARNL(1;130)=Q8VMASK(EPS(1;130),CEVARNL(1;130),BT(1;130);
       1   CEVARNL(1;130))
        CDBNL(1;130)=VALOG10(CEVARNL(1;130);CDBNL(1;130))
        CDBNL(1;130)=10.0*CDBNL(1;130)
    110 BT(1;130)=(SERRNL(1;130).GT.CS(1;130))
        TEMP(1;130)=Q8VMASK(ONE(1;130),ZERO(1;130),BT(1;130);
       1   TEMP(1;130))
        CCSNL(1;130)=VAVG(CCSNL(1;130),TEMP(1;130),K;CCSNL(1;130))
        SERRPL(1;130)=SX1(1;130)-SX1HATPL(1;130)
        CALL MOD2PI(SERRPL)
        CERRPL(1;130)=VAVG(CERRPL(1;130),SERRPL(1;130),K;CERRPL(1;130))
        TEMP(1;130)=SERRPL(1;130)*SERRPL(1;130)
        CESQPL(1;130)=VAVG(CESQPL(1;130),TEMP(1;130),K;CESQPL(1;130))
        IF (K.LE.1) GO TO 120
        CEVARPL(1;130)=CESQPL(1;130)-CERRPL(1;130)*CERRPL(1;130)
        BT(1;130)=(CEVARPL(1;130).LE.EPS(1;130))
```

```
      CEVARPL(1;130)=Q8VMASK(EPS(1;130),CEVARPL(1;130),BT(1;130);
     1   CEVARPL(1;130))
      CDBPL(1;130)=VALOG10(CEVARPL(1;130);CDBPL(1;130))
      CDBPL(1;130)=10.0*CDBPL(1;130)
  120 BT(1;130)=(SERRPL(1;130).GT.CS(1;130))
      TEMP(1;130)=Q8VMASK(ONE(1;130),ZERO(1;130),BT(1;130);
     1   TEMP(1;130))
      CCSPL(1;130)=VAVG(CCSPL(1;130),TEMP(1;130),K;CCSPL(1;130))
      IF (PATH.AND.(K.EQ.1  )) GO TO 130
      GO TO 200
C     PRINT SAMPLE PATH VARIABLES
  130 WRITE (6,6010)
 6010 FORMAT(1H1,36X,21HSAMPLE PATH VARIABLES/
     1   38X,19H(FIRST SAMPLE PATH)/)
      WRITE (6,6011)
 6011 FORMAT(/18X,9HSX1-     ,6X,
     1   36H           PHASE VARIABLES          /
     2   20X,1HI,13X,11HSX1(I)     ,8X,13HSX1(I+1)     /)
      WRITE(6,6100)((I,SX1(I),SX1(I+1)),I=1,129,2)
      WRITE (6,6012)
 6012 FORMAT(/18X,9HSSHAT-   ,6X,
     1   36H         SIN(SX1) ESTIMATES         /
     2   20X,1HI,13X,11HSSHAT(I)   ,8X,13HSSHAT(I+1)   /)
      WRITE(6,6100)((I,SSHAT(I),SSHAT(I+1)),I=1,129,2)
      WRITE (6,6013)
 6013 FORMAT(/18X,9HSCHAT-    ,6X,
     1   36H         COS(SX1) ESTIMATES         /
     2   20X,1HI,13X,11HSCHAT(I)   ,8X,13HSCHAT(I+1)   /)
      WRITE(6,6100)((I,SCHAT(I),SCHAT(I+1)),I=1,129,2)
      WRITE (6,6014)
 6014 FORMAT(/18X,9HSX1HATNL-    ,6X,
     1   36H           NONLINEAR ESTIMATES      /
     2   20X,1HI,13X,11HSX1HATNL(I)    ,8X,13HSX1HATNL(I+1)      /)
      WRITE(6,6100)((I,SX1HATNL(I),SX1HATNL(I+1)),I=1,129,2)
      WRITE (6,6015)
 6015 FORMAT(/18X,9HSERRNL-    ,6X,
     1   36H           NONLINEAR ERRORS         /
     2   20X,1HI,13X,11HSERRNL(I)     ,8X,13HSERRNL(I+1)    /)
      WRITE(6,6100)((I,SERRNL(I),SERRNL(I+1)),I=1,129,2)
      WRITE (6,6016)
 6016 FORMAT(/18X,9HSPLOSSNL-    ,6X,
     1   36H         NONLINEAR PLOSS FUNCTION     /
     2   20X,1HI,13X,11HSPLJSSNL(I)    ,8X,13HSPLOSSNL(I+1)      /)
      WRITE(6,6100)((I,SPLOSSNL(I),SPLOSSNL(I+1)),I=1,129,2)
      WRITE (6,6017)
      6017 FORMAT(/18X,9HTNLF-        ,6X,
          1   36H     NONLINEAR EXECUTION TIMES          /
          2   20X,1HI,13X,11HTNLF(I)          ,8X,13HTNLF(I+1)          /)
           WRITE(6,6100)((I,TNLF(I),TNLF(I+1)),I=1,129,2)
           WRITE (6,6018)
      6018 FORMAT(/18X,9HSX1HATPL-       ,6X,
          1   36H        PHASE LOCK LOOP ESTIMATES        /
          2   20X,1HI,13X,11HSX1HATPL(I)          ,8X,13HSX1HATPL(I+1)          /)
           WRITE(6,6100)((I,SX1HATPL(I),SX1HATPL(I+1)),I=1,129,2)
           WRITE (6,6019)
      6019 FORMAT(/18X,9HSERRPL-        ,6X,
          1   36H           PHASE LOCK LOOP ERRORS        /
          2   20X,1HI,13X,11HSERRPL(I)          ,8X,13HSERRPL(I+1)          /)
           WRITE(6,6100)((I,SERRPL(I),SERRPL(I+1)),I=1,129,2)
           WRITE (6,6020)
      6020 FORMAT(/18X,9HSP11PL-        ,6X,
          1   36HP11 FROM PHASE LOCK RICATTI EQUATION        /
          2   20X,1HI,13X,11HSP11PL(I)          ,8X,13HSP11PL(I+1)          /)
           WRITE(6,6100)((I,SP11PL(I),SP11PL(I+1)),I=1,129,2)
       200 CONTINUE
C          PRINT CUMULATIVE SAMPLE PATH VARIABLES
           IF (CUMPATH.AND.(NMC.GT.1)) GO TO 310
           GO TO 400
       310 WRITE (6,6030)
      6030 FORMAT(1H1,30X,32HCUMULATIVE SAMPLE PATH VARIABLES /)
           WRITE (6,6031)
      6031 FORMAT(/18X,9HCERRNL-        ,6X,
          1   36H      CUMULATIVE NONLINEAR ERRORS     /
          2   20X,1HI,13X,11HCERRNL(I)   ,8X,13HCERRNL(I+1)  /)
           WRITE (6,6100)((I,CERRNL(I),CERRNL(I+1)),I=1,129,2)
           WRITE (6,6032)
      6032 FORMAT(/18X,9HCERRPL-        ,6X,
          1   36HCUMULATIVE PHASE LOCK ERRORS          /
          2   20X,1HI,13X,11HCERRPL(I)   ,8X,13HCERRPL(I+1)  /)
           WRITE (6,6100)((I,CERRPL(I),CERRPL(I+1)),I=1,129,2)
           WRITE (6,6033)
      6033 FORMAT(/18X,9HCESQNL-        ,6X,
          1   36HCUMULATIVE NONLINEAR SQUARED ERRORS     /
          2   20X,1HI,13X,11HCESQNL(I)   ,8X,13HCESQNL(I+1)  /)
           WRITE (6,6100)((I,CESQNL(I),CESQNL(I+1)),I=1,129,2)
           WRITE (6,6034)
      6034 FORMAT(/18X,9HCESQPL-        ,6X,
          1   36HCUMULATIVE PHASE LOCK SQUARED ERRORS    /
          2   20X,1HI,13X,11HCESQPL(I)   ,8X,13HCESQPL(I+1)  /)
           WRITE (6,6100)((I,CESQPL(I),CESQPL(I+1)),I=1,129,2)
      WRITE (6,6035)
 6035 FORMAT(/18X,9HCEVARNL-     ,6X,
     1   36HCUMULATIVE NONLINEAR ERROR VARIANCE      /
     2   20X,1HI,13X,11HCEVARNL(I)   ,8X,13HCEVARNL(I+1)    /)
      WRITE (6,6100)((I,CEVARNL(I),CEVARNL(I+1)),I=1,129,2)
      WRITE (6,6036)
 6036 FORMAT(/18X,9HCEVARPL-     ,6X,
     1   36HCUMULATIVE PHASE LOCK ERROR VARIANCE      /
     2   20X,1HI,13X,11HCEVARPL(I)   ,8X,13HCEVARPL(I+1)    /)
      WRITE (6,6100)((I,CEVARPL(I),CEVARPL(I+1)),I=1,129,2)
      WRITE (6,6037)
 6037 FORMAT(/18X,9HCDBNL-       ,6X,
     1   36HCUMULATIVE NONLINEAR ERROR DECIBELS      /
     2   20X,1HI,13X,11HCDBNL(I)     ,8X,13HCDBNL(I+1)       /)
      WRITE (6,6100)((I,CDBNL(I),CDBNL(I+1)),I=1,129,2)
      WRITE (6,6038)
 6038 FORMAT(/18X,9HCDBPL-       ,6X,
     1   36HCUMULATIVE PHASE LOCK ERROR DECIBELS      /
     2   20X,1HI,13X,11HCDBPL(I)     ,8X,13HCDBPL(I+1)       /)
      WRITE (6,6100)((I,CDBPL(I),CDBPL(I+1)),I=1,129,2)
      WRITE (6,6039)
 6039 FORMAT(/18X,9HCCSNL-       ,6X,
     1   36HCUMULATIVE NONLINEAR CYCLE SLIPS            /
     2   20X,1HI,13X,11HCCSNL(I)     ,8X,13HCCSNL(I+1)    /  /)
      WRITE (6,6100)((I,CCSNL(I),CCSNL(I+1)),I=1,129,2)
      WRITE (6,6040)
 6040 FORMAT(/18X,9HCCSPL-       ,6X,
     1   36HCUMULATIVE PHASE LOCK CYCLE SLIPS      /
     2   20X,1HI,13X,11HCCSPL(I)     ,8X,13HCCSPL(I+1)       /)
      WRITE (6,6100)((I,CCSPL(I),CCSPL(I+1)),I=1,129,2)
C     COMPUTE THE MONTE CARLO AVERGES
  400 IF (NSAMP.LE.30) GO TO 10
      I=NSAMP-30
      T=I
      XERRNL=Q8SSUM(CERRNL(31;I))/T
      XESQNL=Q8SSUM(CESQNL(31;I))/T
      XCSNL=Q8SSUM(CCSNL(31;I))/T
      XERRPL=Q8SSUM(CERRPL(31;I))/T
      XESQPL=Q8SSUM(CESQPL(31;I))/T
      XCSPL=Q8SSUM(CCSPL(31;I))/T
      XEVARNL=XESQNL-XERRNL*XERRNL
      XDBNL=DBEPS
      IF (XEVARNL.GT.EPS(1)) XDBNL=10.0*ALOG10(XEVARNL)
      XEVARPL=XESQPL-XERRPL*XERRPL
      XDBPL=DBEPS
```

```
FORTRAN R1.2 CYCLE 115P2     O=B                SOURCE LISTING            13.39 HRS. 31MAY7
                IF (XEVARPL.GT.EPS(1)) XDBPL=10.*ALOG10(XEVARPL)
        C       PRINT THE MONTE CARLO AVERAGES
                WRITE (6,6050) NMC,XERRNL,XERRPL,XESQNL,XESQPL,XEVARNL,
             1     XEVARPL,XDBNL,XDBPL,XCSNL,XCSPL
          6050 FORMAT(1H1,25X,30HMONTE CARLO SUMMARY STATISTICS//
             1     32X,1H(,I4,14H SAMPLE PATHS)//
             2     36X,16HNONLINEAR FILTER,5X,15HPHASE LOCK LOOP//
             3     14X,14H***VARIABLE***,7X,12H***VALUES***,9X,
             4     12H***VALUES***//14X,13HAVERAGE ERROR,9X,
             5     E15.8,6X,E15.8/
             6     10X,21HAVERAGE SQUARED ERROR,5X,E15.8,6X,E15.8/
             7     14X,14HERROR VARIANCE,8X,E15.8,6X,E15.8        /
             8     14X,14HVARIANCE IN DB,8X,E15.8,6X,E15.8/
             9     11X,19HAVERAGE CYCLE SLIPS,6X,E15.8,6X,E15.8)
          6100 FORMAT(20X,I4,8X,E15.8,4X,E15.8)
                GO TO 10
           500 STOP
                END
   NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2     O=B                SOURCE LISTING          13.39 HRS. 31MAY7
      SUBROUTINE VPRUP(A,I)
      REAL A(16640)
      INTEGER I
      INTEGER MD1,MD2,MD3,MD4,MD5,MD6,MD7,
     1   MD8,MD9,MD10,MD11,MD12,MD13,MD14,MD15,MD16
      COMMON /CPROP/ MD1,MD2,MD3,MD4,MD5,MD6,MD7,
     1   MD8,MD9,MD10,MD11,MD12,MD13,MD14,MD15,MD16
      IF (I.GT.0) GO TO 10
      A(MD2;MD1)=A(1;MD1)
      A(MD4;MD3)=A(1;MD3)
   10 A(MD6;MD5)=A(1;MD5)
      A(MD8;MD7)=A(1;MD7)
      A(MD10;MD9)=A(1;MD9)
      A(MD12;MD11)=A(1;MD11)
      A(MD14;MD13)=A(1;MD13)
      A(MD16;MD15)=A(1;MD15)
      RETURN
      END
  NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2     O=B            SOURCE LISTING           13.39 HRS. 31MAY7
              SUBROUTINE MOD2PI(A)
              REAL A(130),X(130),Y(130)
              BIT BT(130)
              COMMON /PROB/ TWOPI,PI
              X(1;130)=PI
           10 BT(1;130)=(A(1;130).GT.X(1;130))
              IF (Q8SCNT(BT(1;130)).EQ.0) GO TO 20
              Y(1;130)=A(1;130)-TWOPI
              A(1;130)=Q8VMASK(Y(1;130),A(1;130),BT(1;130);A(1;130))
              GO TO 10
           20 X(1;130)=-PI
           30 BT(1;130)=(A(1;130).LT.X(1;130))
              IF (Q8SCNT(BT(1;130)).EQ.0) RETURN
              Y(1;130)=A(1;130)+TWOPI
              A(1;130)=Q8VMASK(Y(1;130),A(1;130),BT(1;130);A(1;130))
              GO TO 30
              END
   NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2    U=B          SOURCE LISTING          13.39 HRS. 31MAY7
00001          REAL FUNCTION VAVG(AV,X,I;*)
00002          DESCRIPTOR AV,X,VAVG
00003          INTEGER I
00004          REAL XI
C0005          XI=I
00906          VAVG =((XI-1.)*AV+X)/XI
00007          RETURN
00008          END
   NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2     O=B              SOURCE LISTING          13.39 HRS. 31MAY7
           REAL FUNCTION RNF(INIT)
           INTEGER INIT,K,KD,M1,M2,N1,N2,NT,MP
           REAL T1,T2
           COMMON /RNUM/ K,KD,M1,M2,N1,N2,NT,MP,T1,T2
           IF (INIT.EQ.0) GO TO 10
           N1=244734
           N2=158551
           N1=102943
           N2=165617
           M1=244734
           M2=158551
           T1=2.**(-18)
           T2=2.**(-36)
           MP=2**18
        10 K=M2*N2
           KD=K/MP
           NT=K-KD*MP
           K=M1*N2+M2*N1+KD
           KD=K/MP
           N1=K-KD*MP
           N2=NT
           RNF=N1*T1+N2*T2
           RETURN
           END
   NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2     O=B               SOURCE LISTING             13.39 HRS. 31MAY7
      REAL FUNCTION GAUSS(INIT,SD,XM)
      INTEGER INIT,I,J
      REAL SD,XM,TWOPI,X,XR1,XR2
      COMMON /GNUM/ I,J,XR2
      COMMON /PROB/ TWOPI
      IF (INIT.EQ.C) GO TO 10
      I=1
      J=1
   10 IF (J.NE.1) GO TO 20
      J=2
      XR1=RNF(I)
      I=0
      XR2=RNF(0)
      X=SQRT(ABS(-2.*ALOG(XR1)))
      XR2=TWOPI*XR2
      XR1=X*SIN(XR2)
      XR2=X*COS(XR2)
      GAUSS=XR1*SD+XM
      RETURN
   20 J=1
      GAUSS=XR2*SD+XM
      RETURN
      END
  NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2      O=B            SOURCE LISTING            13.39 HRS. 31MAY
      SUBROUTINE STATE(MC,SAMP,X1,Z1,Z2)
      INTEGER MC,SAMP,INIT
      REAL X1,Z1,Z2,DEV1,DEV2,DEV3,DEV4,X2
      COMMON /STA/ DEV1,DEV2,DEV3,DEV4,INIT
      COMMON /PROB/ TWOPI,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,A11,A22,
     1   CONST,DEL,FTC,PIDEL,P110,RDEL,RX,QC,Q22,MD,ND
      IF (MC.NE.0) GO TO 10
      DEV1=SQRT(A11)
      DEV2=SQRT(A22)
      DEV3=SQRT(RDEL)
      DEV4=SQRT(Q22)
      INIT=1
      RETURN
   10 IF (SAMP.GT.0) GO TO 20
      X1=GAUSS(INIT,DEV1,Y1EST)
      INIT=0
      X2=GAUSS(0,DEV2,Y2EST)
      RETURN
   20 IF (SAMP.LE.1) GO TO 30
      X1=X1+X2*DEL
      X2=GAUSS(0,DEV4,X2)
   30 Z1=GAUSS(0,DEV3,COS(X1))
      Z2=GAUSS(0,DEV3,SIN(X1))
      RETURN
      END
  NO ERRORS
```

```
FORTRAN R1.2 CYCLE 115P2     U=B          SOURCE LISTING            13.39 HRS. 31MAY7
      SUBROUTINE PLL(MC,SAMP,Z1,Z2,X1HAT,PF11)
      INTEGER MC,SAMP
      REAL Z1,Z2,X1HAT,PF11,DEN,SINF1,COSF1,
     1   X2HAT,PD11,PD12,PD22,PN11,PN12,PN22
      COMMON /PLD/ DEN,X2HAT,PN11,PN12,PN22
      COMMON /PROB/ TWOPI,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,A11,A22,
     1   CONST,DEL,FTC,PIDEL,P110,RDEL,RX,QG,Q22
      IF (MC.LE.0) RETURN
      IF (SAMP.LE.0) GO TO 20
      IF (SAMP.LE.1) GO TO 10
      X1HAT=X1HAT+DEL*X2HAT
      PD11=(RDEL*(PN11+2.0*PN12*DEL)-PN12*PN12
     1   *DEL*DEL)*DEN+PN22*DEL*DEL
      PD12=PN12*(RDEL-PN12*DEL)*DEN+PN22*DEL
      PD22=-PN12*PN12*DEN+PN22+Q22
      PN11=PD11
      PN12=PD12
      PN22=PD22
      DEN=1.0/(PN11+RDEL)
   10 PF11=PN11*RDEL*DEN
      SINF1=SIN(X1HAT)
      COSF1=COS(X1HAT)
      X1HAT=X1HAT+DEN*(-PN11*SINF1*Z1+PN11*COSF1*Z2)
      X2HAT=X2HAT+DEN*(-PN12*SINF1*Z1+PN12*COSF1*Z2)
      RETURN
   20 X1HAT=Y1EST
      X2HAT=Y2EST
      PN11=A11
      PN22=A22
      PN12=0.
      DEN=1.0/(PN11+RDEL)
      RETURN
      END
  NO ERRORS
```

```
      SUBROUTINE NLF(MC,SAMP,Z1,Z2,SHAT,CHAT,TNLF)
C
C-----------------------------------------------------------------------
C
C
C        POINTMASS FILTER FOR TWO DIMENSIONAL PHASE ESTIMATION
C        PROGRAMMED BY KENNETH D. SENNE                  JUNE 24, 1976
C
C
C-----------------------------------------------------------------------
C
      INTEGER MC,SAMP,MD,ND,MND,MND1
      INTEGER MD1,MD2,MD3,MD4,MD5,MD6,MD7,
     1   MD8,MD9,MD10,MD11,MD12,MD13,MD14,MD15,MD16
      REAL Zi,Z2,SHAT,CHAT,TNLF
      INTEGER I,I1,J,J1,J2,K,NC,NT,NTERM,NTERM33
      REAL A11,A22,CL,CNORM,CONST,CR,PI,PIDEL,Q22,T,
     1       TT,Y1EST,Y2EST,TEMP,TEMP1,TEMP2
      REAL S1(64),S2(64),SIGMA(64),PSI(256),A(256)
      REAL IN(8)
      BIT DBT, BT(16640)
      INTEGER DJNM,DJNS
      INTEGER JNS(256),JNM(512)
      REAL COSY(16384),DELJ(16640),JO(16384),JN(16384),JN1(32768),
     1   JNA(32768),SINY(16384),SN1(16384)
      DESCRIPTOR DS1,DS2,DSN1,DSN2,DJN,DJNS,DJNA1,DJNA2,DDELJ,DJN1,
     1   DJNAENT,DJNABNT,DJNANC,DJNAJ1,DJNAJ2,DSINY,DCOSY,DSN1A
      DESCRIPTOR DJNM,DJN1D,DJNA1S,DJN1S,DJNA2S,DBT,DJNA,DJND,DJN1F
      COMMON /PROB/ TWOPI,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,A11,A22,
     1   CONST,DEL,FTC,PIDEL,P110,RDEL,RX,QQ,Q22,MD,ND
      COMMON /NLFC/ NC,NT,NTERM,NTERM33,S1,S2,SIGMA,PSI,A,COSY,DELJ,
     1   JO,JN,JNS,SINY
      COMMON /CPROP/ MDD,MD2,MD3,MD4,MD5,MD6,MD7,
     1   MD8,MD9,MD10,MD11,MD12,MD13,MD14,MD15,MD16
      IF (SAMP.LE.0) GO TO 100
C
C-----------------------------------------------------------------------
C
C        SAMPLE PATH UPDATE TAKES PLACE HERE
C
C
C     SET CLOCK
C
      CALL Q3CLOCKS(T,TT)
C
```

```
C      EVALUATE SENSOR TERMS
C
       DSN1=Z1*DS1
       DSN2=Z2*DS2
       DSN1=DSN1+DSN2
       DSN1=VEXP(DSN1;DSN1)
C
C      PROPAGATE SENSOR TERMS
C
       SN1(33;32)=SN1(1;32)
       SN1(65;64)=SN1(1;64)
       SN1(129;128)=SN1(1;128)
       SN1(257;256)=SN1(1;256)
       SN1(1025;1024)=SN1(1;1024)
       SN1(2049;2048)=SN1(1;2048)
C
C      SCRAMBLE THE JN TO ORDER FOR J(N+1)
C
       CALL Q8VXTOV(X'02',0,DJNM,0,DJND,0,DJN1F)
       CALL Q8VXTOV(X'02',0,DJNS,0,DJN1D,0,DJNA1S)
C
C      FORM THE INTERPOLATED MATRIX
C
       DJN1S=DJNA2S-DJNA1S
       DJN1S=DDELJ*DJN1S
       DJN1S=DJNA1S+DJN1S
C
C      COMPRESS OUT THE LAST ROW OF INTERPOLATED VECTOR
C
       DJNA=Q8VCMPRS(DJN1S,DBT;DJNA)
C
C      COPY THE END COLUMNS
C
       DJNAENT=DJNABNT
C
C      INITIALIZE CONVOLUTION (A(0)=1)
C
       DJN1=DJNANC
C
C      CONVOLUTION LOOP
C
       J1=NC
       J2=NC
       DO 10 I=1,NTERM
       J1=J1+MD
```

```
      J2=J2-MD
      ASSIGN DJNAJ1,JNA(J1;MND)
      ASSIGN DJNAJ2,JNA(J2;MND)
      DJN=DJNAJ1+DJNAJ2
      DJN=A(I)*DJN
   10 DJN1=DJN1+DJN
C
C     MULTIPLY BY SENSOR TERMS
C
      DJN1=DSN1A*DJN1
C
C     GET NORMALIZATION CONSTANT
C
      CNORM=SUMLOG(JN1)
      CNORM=1.0/CNORM
C
C     TRANSFER THE NORMALIZED DENSITY
C
      DJN=CNORM*DJN1
C
C     CUMULATE ESTIMATES
C
      DJNA1=DSINY*DJN
      SHAT=SUMLOG(JNA)
      DJNA1=DCOSY*DJN
      CHAT=SUMLOG(JNA)
C
C     TIMEOUT
C
      CALL Q3CLOCKS(TNLF,TT)
      RETURN
C
C------------------------------------------------------------------------
C
C
C------------------------------------------------------------------------
C
C          SAMPLE PATH INITIALIZATION TAKES PLACE HERE
C
  100 IF (MC.LE.0) GO TO 200
C
C     TRANSFER THE INITIAL DENSITY FOP NEW SAMPLE PATH
C
      JN(1;MND)=JO(1;MND)
      RETURN
```

```
C
C-----------------------------------------------------------------------------
C
C
C-----------------------------------------------------------------------------
C
C          GLOBAL INITIALIZATIONS OCCUR HERE FOR THE ENTIRE RUN
C
C
C      DETERMINE THE NUMBER OF CONVJLUTION POINTS
C
  200 NTERM=(ND/2.)*SQRT(50.*Q22)/PIDEL+0.5
      MD1=MD+1
      MDD=MD
      MD2=MDD+1
      MD3=MDD*2
      MD4=MD3+1
      MD5=MD3*2
      MD6=MD5+1
      MD7=MD5*2
      MD8=MD7+1
      MD9=MD7*2
      MD10=MD9+1
      MD11=MD9*2
      MD12=MD11+1
      MD13=MD11*2
      MD14=MD13+1
      MD15=MD13*2
      MD16=MD15+1
      MND=MD*ND
      M1ND=MD1*ND
      MND1=MND+1
      NTERM33=MD*NTERM
      NT=2*NTERM33
      NC=NTERM33+1
      MND2=2*MND
      ND2=ND*2
C
C      SET UP THE VECTOR DESCRIPTORS FOR THE UPDATE FUNCTIONS
C
      ASSIGN DS1,S1(1;MD)
      ASSIGN DS2,S2(1;MD)
      ASSIGN DSN1,SN1(1;MD)
      ASSIGN DSN2,JNA(1;MD)
      ASSIGN DJN,JN(1;MND)
      ASSIGN DJNM,JNM(1;ND2)
      ASSIGN DJND,JN(1;MD)
      ASSIGN DJN1F,JN1(1;MND2)
      ASSIGN DJNS,JNS(1;ND)
      ASSIGN DJN1D,JN1(1;MD1)
      ASSIGN DJNA1S,JNA(1;M1ND)
      ASSIGN DJN1S,JN1(1;M1ND)
      ASSIGN DJNA2S,JNA(2;M1ND)
      ASSIGN DDELJ,DELJ(1;MND)
      ASSIGN DBT,BT(1;M1ND)
      ASSIGN DJNA,JNA(1;MND)
      ASSIGN DJN1,JN1(1;MND)
      ASSIGN DJNAENT,JNA(MND1;NT)
      ASSIGN DJNABNT,JNA(1;NT)
      ASSIGN DJNANC,JNA(NC;MND)
      ASSIGN DSN1A,SN1(1;MND)
      ASSIGN DSINY,SINY(1;MND)
      ASSIGN DCOSY,COSY(1;MND)
      ASSIGN DJNA1,JNA(1;MND)
C
C     PHASE VARIABLES
C
      DO 210 I=1,MD
      SIGMA(I)=PI*((2.*I-1.)/MD -1.)
      COSY(I)=COS(SIGMA(I))
      SINY(I)=SIN(SIGMA(I))
      S1(I)=COSY(I)/RDEL
  210 S2(I)=SINY(I)/RDEL
      CALL VPROP(SINY,0)
      CALL VPROP(COSY,0)
C
C     PHASE RATE VARIABLES
C
      DO 220 I=1,ND
  220 PSI(I)=PIDEL*((2.*I-1.)/ND  -1.)
C
C     SET UP THE BIT VECTOR
C
      I1=1
      DO 235 I=1,ND
      DO 230 J=1,MD
      BT(I1)=B'1'
  230 I1=I1+1
      BT(I1)=B'0'
  235 I1=I1+1
```

```
C
C      SETUP THE TRANSFER MATRIX
C
       DO 240 J=1,ND
       J1=MOD(ND-1-NTERM+J,ND)*MD*2
       I1=J1+MOD(MD+MD/2+33-(135-NTERM+J)/4,MD)
   240 JNS(J)=I1
C
C      SETUP INTERPOLATION MATRIX
C
       IN(1)=0.875
       IN(2)=0.625
       IN(3)=0.375
       IN(4)=0.125
       IN(5)=IN(1)
       IN(6)=IN(2)
       IN(7)=IN(3)
       IN(8)=IN(4)
       J=MOD(NTERM,4)
       DO 245 I=1,4
       I1=(I-1)*MD1+1
       J1=I+4-J
       T=IN(J1)
   245 DELJ(I1;MD1)=T
C
C      SET UP THE EXPANSION VECTOR
C
       I1=1
       J1=4*MD1
       DO 250 I=1,MD
       DO 250 J=1,J1
       J2=I1+J1
       DELJ(J2)=DELJ(I1)
   250 I1=I1+1
       I1=0
       I2=2*ND
       DO 265 I=2,I2,2
       JNM(I-1)=I1
       JNM(I)=I1
   265 I1=I1+MD
C
C      EVALUATE CONVOLUTION TERMS A(I)
C
       DO 280 I=1,NTERM
       T=I
```

```
      TT=ND
      TEMP=T/TT
      TEMP=CONST*TEMP*TEMP
      A(I)=0.
      IF (TEMP.GT.-47) A(I)=EXP(TEMP)
  280 CONTINUE
C
C     CONSTRUCT THE A PRIORI DENSITY
C
      CNORM=1.0/(TWOPI*SQRT(A11*A22))
      CL=-0.5/A22
      SI=-0.5/A11
      DO 290 I=1,MD
      I1=I
      CR=SIGMA(I)-Y1EST
      CR=CR*CR*SI
      DO 290 J=1,ND
      TEMP=PSI(J)-Y2EST
      JO(I1 )=EXP(TEMP*TEMP*CL+CR)*CNORM
  290 I1=I1+MD
C
C     WRITE OUT PARAMS OF NLF
C
      WRITE (6,6000) MD,ND,MD1,ND
 6000 FORMAT(1H0,26X,27HPOINT MASS NONLINEAR FILTER//1H ,
     1   28X,24HVERSION 2, CODED 6/27/76//1H ,
     2   18X,I3,1HX,I3,25H DENSITIES REPRESENTED BY ,I3,1HX,I3)
      WRITE (6,6001) NTERM,(A(I),I=1,NTERM)
 6001 FORMAT(1H ,33X,7HA(1)-A(,I2,2H) /(1X,5E15.8))
      RETURN
C
C-----------------------------------------------------------------------
C
      END
  NO ERRORS
 FORTRAN R1.2 CYCLE 115P2    U=B            SOURCE LISTING        13.39 HRS. 31MAY7
              FUNCTION SUMLOG(A)
              REAL A(8192), C(4096)
        C
        C   SUMLOG = SUM(A(1), . . ,A(2**NPA))
        C     DOMAIN = 8 .LE. NPA .LE. 13
              NPA = 12
              NA = 2**NPA
              LC = NA/2
              C(1;LC) = A(1;LC)+A(LC+1; LC)
        C      LOOP
          20       LC = LC/2
                   IF(LC .LT. 4) GOTO 50
                   C(1;LC) = C(1;LC)+C(LC+1; LC)
                   GOTO 20
        C      END LOOP
          50  CONTINUE
              SUMLOG =C(1) + C(2) + C(3)+C(4)
              RETURN
              END
   NO ERRORS
```

I - 3 Cray Code

The Cray I from our point of view had the most potential for our problem. However the code development centered on tricks to make the Cray's compiler use the full potential of the machine; in particular to force chaining and efficient use of the available hardware potentialities. It would seem that assembly language coding of this machine should be undertaken in order to effectively use the potential of this machine. We achieved 33 megaflops with the following code. The reader should note that the philosophy that is most useful here is to produce a small number of loops which perform a large number of instructions in the inner loop.

```
C       CRAY FORTRAN COMPILER VERSION 1.05X 02/21/79
C       COMPILATION DATE AND TIME             05/21/79   -    07:46:59
        SUBROUTINE NLF(MC,SAMP,Z1,Z2,SHAT,CHAT,TNLF)
        INTEGER MC,SAMP
        REAL Z1,Z2,SHAT,CHAT,TNLF
           COMMON /LCH1/ T20A, T25A, T30A, T40A,T60A,T70A,T90A
        INTEGER I,I1,J,K,NC,NTERM,NSIZE
        INTEGER JNS(128)
        REAL ALP110,A11,A22,CL,CNORM,CONST,CR,DEL,DELF,FTC,PI,PIDEL,P110,
     1     QQ,Q22,Q22C,RDEL,RX,SI,T,TEMP,TT,T,OPT,Y1EST,Y2EST
        REAL COSY(32),SINY(32),SN1(32),S1(32),S2(32),SIGMA(32),TROW(32)
        REAL A(10),DELJ(128),PSI(128),V1(128),V2(128)
        REAL JN(33,129),JN1(33,149),J0(33,128)
        COMMON /PROB/ TWOPI,PI,ALP110,DELF,Q22C,Y1EST,Y2EST,
     1     A11,A22,CONST,DEL,FTC,PIDEL,P110,RDEL,RX,QQ,Q22
        COMMON /NLFC/ NC,NT,NTERM,S1,S2,SIGMA,PST,A,COSY,
     1     DELJ,J0,JN,SINY

C       IF SAMP NOT POSITIVE THEN REINITIALIZE

        IF (SAMP.LE.0) CO TO 100

C       THE FOLLOWING CONSTITUTES  THE TIME SEGMENT

C       SET CLOCK

        T=SECOND(I)

C       EVALUATE SENSOR TERMS

        DO 10 I=1,32
        V1(I)=Z1*S1(I)+Z2*S2(I)
C       *** NEXT ONE OUT FOR CRAY ***
C       IF (V1(I).LT.-115.) V1(I)=-115.
   10   SN1(I)=EXP(V1(I))

C        TRANSFER JN WITH COLUMNS CYCLICALLY ROTATED TO JN1

           T1 = SECOND(1)
        DO 20 J=1,128
CDIR$ IVDEP
        K=129-J
        DO 15 I=1,32
   15   JN(I+32,K)=JN(I,K)
        DO 20 I=1,33
        JN1(I,K+10)=JN(I+JNS(K),K)
  20   CCONTINUE
           T20= SECOND(1) - T1

C       INTERPOLATE IN JN1 BETWEEN ADJACENT ROWS

           T1 = SECOND (1)
        DO 25 I=1,32
        DO 25 J=11,138
   25   JN1(I,J)=JN1(I+1,J)-JN1(I,J))*DELJ(J-10)+JN1(I,J)
           T25= SECOND(1) - T1
```

```
C       EXPAND ENDS OF JN1 BY CYCLICALLY COPYING COLUMNS

        T2=SECOND(1)
        DO 30 J=1,NTERM
        DO 30 I=1,32
        JN1(I,-J+11)=JN1(I,-J+139)
   30   JN1(I,J+138)=JN1(I,J+10)
           T30= SECOND(1) - T1

C       CONVOLUTION IN JN1 TO JN

        T1=SECOND(1)
        DO 40 I=1,32
        DO 40 J=1,128
        JN(I,J)=JN1(I,J+10) + A(1)*(JN1(I,J+9) + JN1(I,J+11))
      1    + A(2)*(JN1(I,J+8) + JN1(I,J+12))
      2    + A(3)*(JN1(I,J+7) + JN1(I,J+13))
      3    + A(4)*(JN1(I,J+6) + JN1(I,J+14))
      4    + A(5)*(JN1(I,J+5) + JN1(I,J+15))
   40   CONTINUE
           T40= SECOND(1) - T1

C       ACCUMULATE ROW SUMS BY COLUMN

        DO 50 I=1,32
   50   TROW(I)=JN(I,1)
        T1=SECOND(1)
        DO 60 J=2,128
        DO 60 I=1,32
   60   TROW(I)=TROW(I)+JN(I,J)
           T60= SECOND(1) - T1

C       COMPUTE ESTIMATES AND NORMALIZATION CONSTANT

        V1(1)=TROW(1)*SN1(1)
        CNORM=V1(1)
           T1= SECOND(1)
        DO 70 I=2,32
        V1(I)=TROW(I)*SN1(I)
   70   CNORM=CNORM+V1(I)
           T70= SECOND(1) - T1
        SHAT=DOT(V1,1,SINY,1,32)
        CHAT=DOT(V1,1,COSY,1,32)
        CNORM=1./CNORM
        SHAT=SHAT*CNORM
        CHAT=CHAT*CNORM

C       TRANSFER NORMALIZED DENSITY

           T1=SECOND(1)
        DO 90 I=1,32
        TEMP=SN1(I)*CNORM
        DO 90 J=1,128
        JN(I,J)=TEMP*JN(I,J)
   90   CONTINUE
           T90= SECOND(1) - T1
        TNLF=SECOND(TT)-T
           PRINT 1234,T20,T25,T30,T40,T60,T70,T90
 1234      FORMAT(" 20, 25, 30, 40, 60, 70, 90 ",7F12.8)
```

```
           T20A=T20A+T20
           T25A=T25A+T25
           T30A=T30A+T30
           T40A=T40A+T40
           T60A=T60A+T60
           T70A=T70A+T70
           T90A=T90A+T90
C          THE VARIABLES ABOVE ARE INITIALIZED TO ZERO BY THE LOADER SINCE
C             THEY ARE IN LABELLED COMMON (SOPPY BUT SHOULD BE OK).

C         TIMEOUT

C         THIS ENDS THE TIMED SEGMENT

          RETURN

C * * * * NOTHING BELOW THIS POINT REQUIRES VECTORIZATION * * * *

C         IF MC NOT POSITIVE THEN GLOBAL INITIALIZE

  100     IF (MC.LE.0) GO TO 200

C         SAMPLE PATH INITIALIZATION

          DO 110 I=1,32
          DO 110 J=1,128
  110     JN(I,J)=J0(I,J)
          RETURN

C         GLOBAL INITIALIZATIONS FOR NONLINEAR FILTER

  200     NSIZE=10
          NTERM=64.0*SQRT(50.*Q22)/PIDEL+0.5
          IF(NTERM.GT.NSIZE) NTERM=NSIZE

C         PHASE VARIABLES

          DO 210 I=1,32
          SIGMA(I)=PI*((2.*I-1.)/32.-1.)
          COSY(I)=COS(SIGMA(I))
          SINY(I)=SIN(SIGMA(I))
          S1(I)=COSY(I)/RDEL
  210     S2(I)=SINY(I)/RDEL

C         PHASE RATE VARIABLES

          DO 220 I=1,128
  220     PSI(I)=PIDEL*((2.*I-1.)/128.-1.)

C         SETUP THE TRANSFER MATRIX

          DO 240 I=1,128
  240     JNS(J)=MOD(47-(J-1)/4,32)
```

```
C     SETUP THE INTERPOLATION VECTOR

      DELJ(1)=0.875
      DELJ(2)=0.625
      DELJ(3)=0.375
      DELJ(4)=0.125
      DO 250 I=5,125,4
      DELJ(I)=DELJ(I-4)
      DELJ(I+1)=DELJ(I-3)
      DELJ(I+2)=DELJ(I-2)
  250 DELJ(I+3)=DELJ(I-1)

C     EVALUATE CONVOLUTION TERMS A(I)

      DO 280 I=1,NTERM
      TEMP=I/128.
      TEMP=CONST*TEMP*TEMP
      A(I)=0.
      IF (TEMP.GT.-47.) A(I)=EXP(TEMP)
  280 CONTINUE

C     CONSTRUCT THE A PRIORI DENSITY

      CNORM=1.0/(TWOPI*SQRT(A11*A22))
      CL=-0.5/A22
      SI=-0.5/A11
      DO 290 I=1,32
      CR=SIGMMA(I)-Y1EST
      CR=CR*CR*SI
      DO 290 J=1,128
      TEMP=PSI(J)-Y2EST
      TEMP=TEMP*TEMP*CL*CR
C     *** NEXT TWO OUT FOR CRAY ***
C     J0(I,J)=0.
C     IF (TEMP.GT.-115.) J0(I,J)=EXP(TEMP)*CNORM
C     *** NEXT ONE IN FOR CRAY ***
      J0(I,J)=EXP(TEMP)*CNORM
  290 CONTINUE
      RETURN
      END


>
```

II - 1 CDC 7600 Code for the 3-D Problem

The three dimensional phase demodulation program code was developed simultaneously for the 7600 and the Star 100 in order to have a check on each. The 7600 was not effective on this program as with level 433 the optimizing compiler did not produce runable code, the extended memory address, calculations failed. When the Opt=1 compiler was used, times of 20 to 30 times slower than the Star resulted, while on the 2D problem the 7600 was only 5 times slower than Star. The compiler failure was submitted to Control Data as a problem and acknowledged but never solved. The reader could view the 7600 code as the scalar version of the 3D Star Code which follows in II - 2. It is apparent that the 7600 is not as effective as the Star and Cray on problems with large memory requirement, i.e. around 350K.

```
      PROGRAM MAIN(INPUT,OUTPUT,PUNCH,TAPE6=OUTPUT,TAPE5=INPUT)
      DIMENSION XDAT(130,10),Y1(35),Y2(135),EXP33(16,16),YA(35),
     $EI(16),EJ(96),EK(16),
     $EXPON(35),EXOON(17),Y3(17)
      REAL COSY(24576),SINY(24576),SN1(24576),SN2(1536),JN(24576),
     $JN1(26113),JNA(54272),DELJ(26112),S1(16),S2(16),YB(24576),D(2000),
     $YC(16)
      INTEGER JNS(1536),JNF(3072)
C     BIT B3(26112)
      LOGICAL B3( 26112 )
      LEVEL 2,JNA
      LEVEL 2,DELJ, JN1,YB
      LEVEL 2,COSY,SINY,SN1
      LEVEL 2,SN2,JNS
      LEVEL 2,B3
      COMMON/GN/JGAUSS,XZZZ(2)
      COMMON /A/JNA
      COMMON /B/ DELJ,JN1,YB
      COMMON /C/ COSY,SINY,SN1,SN2,JNS,B3
      NAMELIST /INSTR/Y3EST,Q33C,ALF,GAM,NUM3,Y1EST,Y2EST,ALP110,DELF,
     *                    Q22C,NUM1,NUM2,NO2,NO3
      WRITE(6,666)
  666 FORMAT(1H1,5HINPUT)
      READ(5,INSTR)
      WRITE(6,INSTR)
      JGAUSS = 0
      P110 = 10.**(ALP110/10.)
      Q0 = Q22C**(.25)
      RX = (P110/(SQRT(2.0)*Q0))**(4.0/3.0)
      FTC = SQRT(2.0)*RX**(.25)/Q0
      DELT = DELF*FTC
      Q22 = Q22C*DELT
      R11 = RX/DELT
      P220 = P110*SQRT(Q22C/RX)
      R11M1=1./R11
      ALFD = ALF*DELT
      BET = 1.0 - ALFD
      A11 = 10.**((ALP110+GAM)/10.)
      A22 = P220
      P330 = .5*Q33C/ALF
      A33  = 2.0*P330
      Q33 = Q33C*DELT
      DEV1 = SQRT(A11)
      DEV2 = SQRT(A22)
      DEV3 = SQRT(R11)
      DEV4 = SQRT(A33)
      DEVQ2 = SQRT(Q22)
      DEVQ3 = SQRT(Q33)
      KOUNT = 1
      ISAMP = 1
      NSAMP = 0
      SUMP1 = 0.0
      SUMP2 = 0.0
      SUMP3 = 0.0
      DELSQ = DELT**2
      PI = 4.*ATAN(1.)
      PI2 = 2.0*PI
      PIDLT = PI/DELT
      PINV = 1.0/PI
      PI2DLT = 2.0*PIDLT
      U1 = NUM1
      U2 = NUM2
      U3 = NUM3
      IY2 = U2/PI2DLT*SQRT(50.*D22)+.5
      NTERM = IY2
      NTERM1 = NTERM + 1
      NTRM16 = NTERM*16
      NC = NTRM16  + 1
      NT = 2*NTRM16
      NT1536  = NT + 1536
      R55 = .5/R11
      CL = -.5/A22
      CM = -.5/A33
      SI = -.5/A11
C ****************** GRID Y1, Y2 AND YA *********************
      EDG1 = PI/U1
      EDG2 = PIDLT/U2
      DO 40 I = 1,NUM1
         X = I - 1
         X = X/U1
         Y1(I) = -PI + X*PI2 + EDG1
   40    CONTINUE
      DO 50 I = 1,NUM2
         X = I - 1
         X = X/U2
         Y2(I) = -PIDLT + X*PI2DLT + EDG2
   50    CONTINUE
      DO 51 I = 1,IY2
         X = I
         X = X/U2
         Y3(I) = X
   51    CONTINUE
      DO 55 I = 1,NUM1
         COSY(I) = COS(Y1(I))
         SINY(I) = SIN(Y1(I))
   55    CONTINUE
      DO 383 I = 1,16
         S1(I) = COSY(I)/R11
         S2(I) = SINY(I)/R11
  383    CONTINUE
      CALL VPROP(SINY,SINY,16,24576,16)
      CALL VPROP(COSY,COSY,16,24576,16)
   71 XP=.5*SQRT(A33)
      YY3 = (NUM3 - 1.0)/2.0 + 1.0
      IY3 = YY3
      DO 60 I = 1,NUM3
         XX = I - YY3
         YA(I) = Y3EST + XP*XX
         YC(I)=Y3EST+XP*XX
   60    CONTINUE
C ******************** DYNAMIC EXPONENTIALS *********************
      IF (IY2 .EQ. 0)  GO TO 153
      DO 150 I = 1,IY2
      DON=PIDLT*PI2DLT
```

```
         EXDON(I) = EXP(-DON/Q22*(Y3(I)**2))
  150    CONTINUE
      DO 152 I = 1,IY2
         EXPON(I) = EXDON(IY2+1-I)
         EXPON(IY2+1+I) = EXDON(I)
  152    CONTINUE
  153 EXPON(IY2+1) = .5
      IYY = 2*IY2 + 1
      LTERM2 = 0
         DO 720 K = 1,16
      DO 720 I = 1,16
            XNUM = (K - I)*XP + ALFO*(YA(I) - 1.)
            XNUM = -.5/Q33*XNUM**2
            EXP33(I,K) = 0.0
            IF (XNUM .LT. -27.)  GO TO 720
            EXP33(I,K) = EXP(XNUM)
  720       CONTINUE
      LTERM = 0
      LTRM16  = LTERM*16
      LTERM1 = LTERM + 1
      LC = LTRM16  + 1
      NS = NT1536 *LTERM + NC
C   NOT ON ORIGINAL LISTING, BUT ON ICASE REPORT LISTING
      I = 0
      DO 339 K = 1,16
         DO 339 N = 1,16
            DO 339 J = NTERM1,IYY
               I = I + 1
               D(I) = EXPON(J)*EXP33(N,K)
  339          CONTINUE
      ALOSS=A33
      ALOSSO=ALOSS
      X3EST=Y3EST
      X3ESTO=Y3EST
C ****************************  INITIAL DENSITY *********************
      DO 155 I=1,16
         XXX=ST*(Y1(I)-Y1EST)**2
         IF(XXX .LT. -27) GO TO 154
         EI(I)=EXP(XXX)
         GO TO 155
  154    EI(I)=0.0
  155    CONTINUE
      DO 157 J=1,96
         YYY=CL*(Y2(J)-Y2EST)**2
         IF(YYY .LT. -27) GO TO 156
         EJ(J)=EXP(YYY)
         GO TO 157
  156    EJ(J)=0.0
  157    CONTINUE
      DO 159 K=1,16
         ZZZ=CM*(YA(K)-Y3EST)**2
         IF(ZZZ .LT. -27) GO TO 158
         EK(K)=EXP(ZZZ)
         GO TO 159
  158    EK(K)=0.0
  159    CONTINUE
      IJK = 0
      DO 160 K = 1,16
         DO 160 J = 1,96
            DO 160 I = 1,16
               IJK = IJK + 1
               JN(IJK)=EK(K)*EI(I)*EJ(J)
  160          CONTINUE
C     WRITE(6,9998) JN
 9998 FORMAT(1P8E14.6)
C ****************************  INTEGER ARRANGMENT **************
      DO 225 I = 1,26112
         B3(I) = .TRUE.
  225    CONTINUE
      I = 0
      DO 300 K = 1,16
         DO 300 J = 1,96
            I = I + 17
            B3(I) = .FALSE.
  300       CONTINUE
      DO 320 K = 1,3071,2
         JNF(K) = (K - 1)*8
  320    JNF(K+1) = JNF(K)
      I = 0
      J1 = 0
      DO 332 K = 1,16
         DO 332 J = 1,96
            I = I + 1
            I1 = J1 + MOD(23-(J-1)/6,16)
            JNS(I) = I1
            J1 = J1 + 32
  332       CONTINUE
      DO 7332 I = 1,17
         DELJ(I) = 11./12.
         DELJ(I+17) = .75
         DELJ(I+34) = 7./12.
         DELJ(I+51) = 5./12.
         DELJ(I+69) = .25
         DELJ(I+85) = 1./12.
 7332    CONTINUE
C      CALL VPROP1(DELJ) ***********))))))))))))
      CALL VPROP(DELJ,DELJ,102,26010,102)
   11 CONTINUE
      KOUNT = 1
      CALL GAUSS(JSEED,DEV1,Y1EST,X1)
      XDAT(KOUNT,1) = X1
      CALL GAUSS(JSEED,DEV2,Y2EST,X2)
      CALL GAUSS(JSEED,DEV4,Y3EST,X3)
      ACOS=EXP(X3-1.)*COS(X1)
      ASIN=EXP(X3-1.)*SIN(X1)
      CALL GAUSS(JSEED,DEV3,ACOS,Z1)
      CALL GAUSS(JSEED,DEV3,ASIN,Z2)
      GO TO 470
  450 CONTINUE
      X1 = X1 + X2*DELT
      XDAT(KOUNT,1) = X1
      CALL GAUSS(JSEED,DEVO2,X2,X2)
      X3 = X3*BET + ALFO
      CALL GAUSS(JSEED,DEVO3,X3,X3)
      XDAT(KOUNT,5)=X3
      ACOS=EXP(X3-1.)*COS(X1)
      ASIN=EXP(X3-1.)*SIN(X1)
      CALL GAUSS(JSEED,DEV3,ACOS,Z1)
      CALL GAUSS(JSEED,DEV3,ASIN,Z2)
      XP = .5*SQRT(ALOSS)
C     XP=.5*AMAX1(.001,SQRT(ALOSS))
      XPO=.5*SQRT(ALOSSO)
C     LTERM = 0
      DO 600 I = 1,NUM3
         XX = I - YY3
         YA(I)=X3ESTO+XX*XPO
         YC(I)=X3EST+XX*XP
  600    CONTINUE
         DO 730 K = 1,16
      DO 730 I = 1,16
            XNUM=ALFO*(YA(I)-1.)+YC(K)-YA(I)
            XNUM = -.5/D33*XNUM**2
            EXP33(I,K) = 0.0
            IF (XNUM .LT. -27.) GO TO 730
            EXP33(I,K) = EXP(XNUM)
  730       CONTINUE
      I = 0
      DO 340 K = 1,16
         DO 340 N = 1,16
            TEMP = EXP33(N,K)
            DO 340 J = NTERM1,IYY
            I = I + 1
  340       D(I) = EXPON(J)*TEMP
  470 CONTINUE
C ************************  SENSOR FUNCTION ****************
      CALL SECOND(TIMEIN)
      DO 7500 I = 1,16
      R11TZ1=Z1*R11M1
      R11TZ2=Z2*R11M1
      S1(I)=R11TZ1*COS(Y1(I))
      S2(I)=R11TZ2*SIN(Y1(I))
 7500    S1(I) = S1(I) + S2(I)
      J = 0
      DO 500 K = 1,16
         DO 860 KK = 1,16
            S2(KK)=S1(KK)*EXP(YA(K)-1.)
            SN2(KK) = EXP(S2(KK))
  860       CONTINUE
C ********** CALL VPROP(SN2,0) ))))))))))))))))))
      CALL VPROP(SN2,SN2,16,1536,16)
         DO 870 KK = 1,1536
            SN1(KK+J) = SN2(KK)
  870       CONTINUE
         DO 880 KK = 1,16
            S2(KK)=-S55*EXP(YA(K)-1.)*EXP(YA(K)-1.)
            SN2(KK) = EXP(S2(KK))
  880       CONTINUE
C ************ CALL VPROP(SN2,0) )))))))))))))))
      CALL VPROP(SN2,SN2,16,1536,16)
         DO 890 KK = 1,1536
            SN1(KK+J) = SN1(KK+J)*SN2(KK)
 690      CONTINUE
      J = J + 1536
 500    CONTINUE
C     ***************   MAIN LOOP STARTS  ********************
C ****** CALL OBVXTOV(X*C2,0,KJNF,0,DB,0,0 JNA)
    DO 707 KK = 1,24576
    JN(KK)=JN(KK)*SN1(KK)
 707    CONTINUE
    J = 1
    DO 991 K = 1,3072
      DO 992 I = 1,16
        JNA(J) = JN(JNF(K)+I)
        J = J + 1
 992      CONTINUE
 991    CONTINUE
C ****** CALL       **( *      KJNS   DC         1)
    J = 1
    DO 993 K = 1,1536
      DO 994 I = 1,17
        JN1(J) = JNA(JNS(K)+I)
        J = J + 1
 994      CONTINUE
 993    CONTINUE
    JN1(26113) = 0.0
    DO 900 KK = 1,26112
      JNA(KK) = JN1(KK+1) - JN1(KK)
      JNA(KK) = DELJ(KK)*JNA(KK)
      JN1(KK) = JN1(KK) + JNA(KK)
 900    CONTINUE
    CALL  PRVEC(4HJN1  ,  JN1)
    I7 = 1
    DO 902 KK = 1,26112
      IF (.NOT. B3(KK)) GO TO 902
      JNA(I7) = JN1(KK)
      JN1(I7) = JNA(I7)
      I7 = I7 + 1
 902    CONTINUE
C   WRITE(6,9998) JN1
    J = 0
    I = 0
    DO 510 K = 1,16
      N = I + NTPM16
      DO 511 KK = 1,1536
        JNA(KK+N) = JN1(KK+J)
 511      CONTINUE
      DO 512 KK = 1,NTPM16
        JNA(KK+I) = JNA(KK+I+1536)
 512      CONTINUE
      DO 513 KK = 1,NTPM16
        JNA(N+1536+KK) = JN1(KK+J)
 513      CONTINUE
      J = J + 1536
      I = I + NT1536
 510    CONTINUE
    CALL  PRVEC(4HJNA  ,  JNA)
    N = 0
    I1 = 0
```

```
      JK = 1
      DO 700 I = 1,16
      I1=0
         DO 701 KK = 1,1536
            JN1(N+KK) = 0.0
  701       CONTINUE
         DO 690 K = 1,16
            J1 = NS + I1 - 1
            J2 = NS + I1 - 1
            DO 680 J = 1,NTERM1
               DO 703 KK = 1,1536
                  JN(KK) = JNA(KK+J1) + JNA(KK+J2)
                  JN(KK) = JN(KK)*D(JK)
                  JN1(N+KK) = JN1(N+KK) + JN(KK)
  703             CONTINUE
               JK = JK + 1
               J1 = J1 + 16
               J2 = J2 - 16
  680          CONTINUE
  690 I1=I1+NT1536
         N = N + 1536
  700    CONTINUE
      CALL  PRVEC(4HJN1T , JN1)
C     WRITE(6,9998) JN1
      CNORM = SUMLOG(JN1,24576)
      IF (CNORM .LT. 1.E-20) CNORM = 1.0
      CNORM = 1./CNORM
      DO 710 KK = 1,24576
         JN (KK) = CNORM*JN1(KK)
         JNA(KK) = COSY(KK) * JN(KK)
  710    CONTINUE
      CHAT = SUMLOG(JNA,24576)
      DO 711 KK = 1,24576
         JNA(KK) = SINY(KK) * JN(KK)
  711    CONTINUE
      SHAT = SUMLOG(JNA,24576)
      CXHAT = ATAN2(SHAT,CHAT)
      J = 0
      DO 343 K = 1,16
         DO 770 KK = 1,1536
            YB(KK+J)=YC(K)
  770       CONTINUE
         J = J + 1536
  343    CONTINUE
      DO 771 KK = 1,24576
         JNA(KK) = YB(KK)*JN(KK)
  771    CONTINUE
         X3ESTO=X3EST
      X3EST = SUMLOG(JNA,24576)
      DO 773 KK = 1,24576
         JNA(KK) = JNA(KK)*YB(KK)
  773    CONTINUE
         ALOSSO=ALOSS
      ALOSS = SUMLOG(JNA,24576)
      ALOSS = ALOSS - X3EST*X3EST
      ALOSS=AMAX1(ALOSS,1.E-18)
C     ********************  MAIN LOOP ENDS  ****************
```

```
      CALL SECOND(TIMOUT)
      TNLF = TIMOUT - TIMEIN
      WRITE(6,201) KOUNT,X1,X2,X3,SHAT,CHAT,CXHAT,X3EST,ALOSS
  201 FORMAT( I5,1X,1P3E14.6,4X,1P2E14.6,4X,1P3E14.6  )
      WRITE(6,8880) TNLF
 8880 FORMAT( 1PE12.6 )
      IF (KOUNT .EQ. NO2 )  GO TO 505
      XDAT(KOUNT,2)=CXHAT
      XDAT(KOUNT,3)=ALOSS
      XDAT(KOUNT,4)=X3EST
      KOUNT = KOUNT + 1
      GO TO 450
  505 CONTINUE
      SUMP = 0.0
      SUMC = 0.0
      XDAT(KOUNT,2) = CXHAT
      XDAT(KOUNT,3) = ALOSS
      XDAT(KOUNT,4) = X3EST
      DO 1501 I = 31,NO2
         XD = ABS(XDAT(I,1) - XDAT(I,2))
 1498    CONTINUE
         IF (XD .GT. PI)  GO TO 1499
         GO TO 1500
 1499    XD = XD - PI2
         GO TO 1498
 1500    SUMP = SUMP + XD*XD
 1501    CONTINUE
      H = NO2 - 30
      SUMP = SUMP/H
      XNSAMP = NSAMP
      XAA = XNSAMP + 1.0
      SUMP1 = (SUMP + XNSAMP*SUMP1)/XAA
      DSUMP1 = ALOG10(SUMP1)*10.0
      DO 1601 I = 31,NO2
         XD=ABS(XDAT(I,5)-XDAT(I,4))
 1698    CONTINUE
         IF (XD .GT. PI)  GO TO 1699
         GO TO 1700
 1699    XD = XD - PI2
         GO TO 1698
 1700    SUMC = SUMC + XD*XD
 1601    CONTINUE
      SUMC = SUMC/H
      SUMP2 = (SUMC + XNSAMP*SUMP2)/XAA
      DSUMP2 = ALOG10(SUMP2)*10.0
      WRITE(6,1511) NSAMP,SUMP1,DSUMP1,SUMP2,DSUMP2
      NSAMP = NSAMP + 1
      IF (ISAMP .EQ. NO3)  GO TO 2200
      ISAMP = ISAMP + 1
      GO TO 71
 2200 CONTINUE
      STOP
 8889 FORMAT(1H ,2(5E14.7,/,1H ))
 9671 FORMAT(1H , I2)
 1511 FORMAT( I10,1P4E14.6,1H  )
      END
```

```
    SUBROUTINE VPROP        76/76    OPT=1                                      FTN 4.6+433A

   1                   SUBROUTINE VPROP(FROM,TO,IGO,IEND,INC)
                       DIMENSION FROM(1),TO(1)
                       LEVEL 2,FROM,TO
                       DO 10 I = IGO,IEND,INC
   5                      DO 20  J = 1,INC
                             TO(I+J) = FROM(J)
                    20       CONTINUE
                 C
                    10    CONTINUE
  10                   RETURN
                 C
                       END



    SYMBOLIC REFERENCE MAP (R=3)

NTRY POINTS       DEF LINE      REFERENCES
   3  VPROP            1          10

ARIABLES       SN   TYPE              RELOCATION
   0  FROM          REAL        ARRAY      F.P.        REFS          2          3    6
  36  I             INTEGER                            REFS          6    DEFINED    4
   0  IEND          INTEGER                F.P.        REFS          4    DEFINED    1
   0  IGO           INTEGER                F.P.        REFS          4    DEFINED    1
   0  INC           INTEGER                F.P.        REFS          4          5  DEFINE
  37  J             INTEGER                            REFS        2*6    DEFINED    5
   0  TO            REAL        ARRAY      F.P.        REFS          2          3  DEFINE

TATEMENT LABELS                   DEF LINE   REFERENCES
   0  10                              9          4
   0  20                              7          5

OOPS  LABEL     INDEX       FROM-TO     LENGTH     PROPERTIES
  20  10      * I             4  9        15B                  NOT INNER
  25  20        J             5  7         4B      INSTACK

TATISTICS
 PROGRAM LENGTH                  53B      43
```

```
      FUNCTION SUMLOG      76/76    OPT=1                                        FTN 4.6+433A

   1                    FUNCTION SUMLOG(A,N)
                        DIMENSION A(1)
                        LEVEL 2,A
                        SUMLOG = 0.0
   5                    DO 10 I = 1,N
                           SUMLOG = SUMLOG + A(I)
                  10       CONTINUE
               C
                        RETURN
  10           C
                        END



     SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS        DEF LINE       REFERENCES
    4  SUMLOG           1             9

VARIABLES       SN  TYPE                RELOCATION
    0  A            REAL          ARRAY      F.P.        REFS         2         3
   20  I            INTEGER                              REFS         6    DEFINED
    0  N            INTEGER                  F.P.        REFS         5    DEFINED
   17  SUMLOG       REAL                                 REFS         6    DEFINED

STATEMENT LABELS                   DEF LINE    REFERENCES
    0  10                              7            5

LOOPS  LABEL     INDEX        FROM-TO      LENGTH     PROPERTIES
   12  10        I               5 7           4B      INSTACK

STATISTICS
  PROGRAM LENGTH                      21B      17
  1                     SUBROUTINE GAUSS(JS,SD,XM,X)
                        DIMENSION NST(3)
                        COMMON /RN/ N1,N2,N3,MC,T1,T2,T3
                        COMMON /GN/ J, XR(2)
  5                     IF (J) 10, 10, 20
                     10 J = 2
                        TWOPI = 8.*ATAN(1.)
                        NST(1) = 1608
                        NST(2) = 2029
 10                     NST(3) = 1297
                        XR(1) = RNNF(NST,1)
                        GO TO 35
                     20 GO TO (30,40), J
                     30 J = 2
 15                     XR(1) = RNNF(NST,0)
                     35 XR(2) = RNNF(NST,0)
                        X1 = SQRT(ABS(-2.*ALOG(XR(1))))
                        XR(2) = TWOPI*XR(2)
                        XR(1) = X1*SIN(XR(2))
 20                     XR(2) = X1*COS(XR(2))
                        X = XR(1)*SD + XM
                        RETURN
                     40 J = 1
                        X = XR(2)*SD+XM
 25                     RETURN
                        END
```

```
CARD NR. SEVERITY  DETAILS    DIAGNOSIS OF PROBLEM

    13      I                AN IF STATEMENT MAY BE MORE EFFICIENT THAN A 2 OR 3 BRAN
```

```
     SYMBOLIC REFERENCE MAP (R=3)

 ENTRY POINTS        DEF LINE        REFERENCES
     3   GAUSS            1            22          25

 VARIABLES        SN   TYPE               RELOCATION
     0   J             INTEGER                GN          REFS          4          5
     0   JS            INTEGER      *UNUSED    F.P.     DEFINED          1
     3   MC            INTEGER                RN          REFS          3
    77   NST           INTEGER       ARRAY                REFS          2         11
                                                            15
     0   N1            INTEGER                RN          REFS          3
     1   N2            INTEGER                RN          REFS          3
     2   N3            INTEGER                RN          REFS          3
     0   SD            REAL                    F.P.       REFS         21         24     DEFI
    75   TWOPI         REAL                               REFS         18      DEFINED
     4   T1            REAL                   RN          REFS          3
     5   T2            REAL                   RN          REFS          3
     6   T3            REAL                   RN          REFS          3
```

```
      1                    FUNCTION RNNF(NS,MODE)
                           DIMENSION NC(3),NS(3)
                           COMMON /PN/ N1,N2,N3,MP,T1,T2,T3
                           DATA M1,M2,M3 /3823,4036,2903/
      5                    IF (MODE) 10, 100, 10
                        10 N1 = NS(1)
                           N2 = NS(2)
                           N3 = NS(3)
                           T1 = 2.**(-12)
     10                    T2 = 2.**(-24)
                           T3 = 2.**(-36)
                           MP = 2**12
                       100 K = N3*M3
                           KD = K / MP
     15                    NC1 = K - KD*MP
                           K = N3*M2 + N2*M3 + KD
                           KD = K / MP
                           NC2 = K - KD*MP
                           K = N3*M1 + N2*M2 + N1*M3 + KD
     20                    KD = K / MP
                           NC3 = K - KD*MP
                           N1 = NC3
                           N2 = NC2
                           N3 = NC1
     25                    XN1 = N1
                           XN2 = N2
                           XN3 = N3
                           RNNF = XN1*T1 + XN2*T2 + XN3*T3
                           RETURN
     30                    END
```

```
      SYMBOLIC REFERENCE MAP (P=3)

ENTRY POINTS       DEF LINE      REFERENCES
    4  RNNF           1            29

VARIABLES       SN  TYPE               RELOCATION
   63  K            INTEGER                              REFS        14         15        1
                                                      DEFINED        13         16        1
   64  KD           INTEGER                              REFS        15         16        1
                                                      DEFINED        14         17        2
    0  MODE         INTEGER                 F.P.         REFS         5    DEFINED
    3  MP           INTEGER                PN            REFS         3         14        1
                                                      DEFINED        12
   56  M1           INTEGER                              REFS        19    DEFINED
   57  M2           INTEGER                              REFS        16         19    DEFIN
   60  M3           INTEGER                              REFS        13         16        1
   73  NC           INTEGER    *UNDEF                    REFS         2
   65  NC1          INTEGER                              REFS        24    DEFINED        1
   66  NC2          INTEGER                              REFS        23    DEFINED        1
   67  NC3          INTEGER                              REFS        22    DEFINED        2
    0  NS           INTEGER    ARRAY        F.P.         REFS         2          6
    0  N1           INTEGER                PN            REFS         3         19        2
```

```
     SUBROUTINE PRVEC        76/76    OPT=1                                    FTN 4.6+430

  1                  SUBROUTINE  PRVEC(LABEL,VEC)
              C   PRINT   SELECTED VECTOR COMPONENTS
                     INTEGER  LABEL,DIM12,COUNT
                     REAL VEC(1)
  5                  LEVEL 2,VEC
                     DATA  DIM12,COUNT/1536, 0/
                     COUNT = COUNT+1
              C      WRITE(6,99) COUNT, LABEL,
              C     *   VEC(DIM12*7 + 1), VEC(DIM12*7+750),
 10           C     *   VEC(DIM12*7+1148)
              C 99   FORMAT(11H PR. ENTRY    ,I5,9H AT ENT.    ,A4,
              C     *   5E14.7)
                     RETURN
                     END




     SYMBOLIC REFERENCE MAP (R=3)

ENTRY POINTS       DEF LINE       REFERENCES
    3  PRVEC             1           13

VARIABLES       SN   TYPE                RELOCATION
   10  COUNT         INTEGER                                    REFS          3           7     DEFINE
    7  DIM12         INTEGER                                    REFS          3     DEFINED          6
    0  LABEL         INTEGER     *UNUSED      F.P.              REFS          3     DEFINED          1
    0  VEC           REAL          ARRAY      F.P.              REFS          4           5     DEFINE

STATISTICS
```

II - 2 CDC Star-100

The code for the 3-D phase demodulation problem was a natural extension of that for the 2-D problem described earlier. The model for the problem is described in [1] . In particular the problem pushed the capacity of Star to its limit as the density now was represented as a 25,000 word vector which because of the algorithm structure required close to the 65,000 word vector limit of the machine. This code was not used for Monte Carlo production runs because of computer time limitations, but as a check on the accuracy of the assembly code for the AP120B code of the next section.

```
FORTRAN R1.3 CYCLE I          BUILT 09/27/78 20:40     SOURCE LISTING
                PROGRAM MAIN(INPUT,OUTPUT,PUNCH,UNIT6=OUTPUT,UNIT5=INPUT,
             X        UNIT10=SEED)
                DIMENSION XDAT(130,10),XHAT(4),Y1(35),Y2(135),
               *EXPON(35),EXDON(17),Y3(17),PNF(3,3)     ,YA(35),EXP33(16,16),
               *Q(3,3),PBAR(3,3),PN(3,3),AN(3),F(3,3),PDUMMY(3,3),PDUMY2(3,3)
                COMMON/GN/JGAUSS,XZZZ(2)
                REAL COSY(24576),SINY(24576),SN1(24576),SN2(1536),JN(24576),
               *JN1(26112),JNA(54272),DELJ(26112),S1(16),S2(16),YB(24576),D(2000)
                REAL YC(16)
                INTEGER  JNS(1536),JNF(3072)
                INTEGER OPSEED,SDNORM,SDWRT,SDRDWR,SDSTAR,SDRUN,SDSAVE,SDREST
                BIT B3(26112)
                DESCRIPTOR DB,KJNF,DJNA,KJNS,DJN1,DC
                DATA  SDNORM,SDWRT,SDRDWR,   SDSTAR,SDRUN,SDSAVE,SDREST
             X       /0,1,2,    1,2,3,4/
                NAMELIST /INSTR/ Y3EST,Q33C,ALF,GAM,NUM3,
             X       Y1EST,Y2EST,ALP110,DELF,Q22C,NUM1,NUM2,NO2,NO3,OPSEED,
             X       IPRIN
                ASSIGN DB,JN(1;16)
                ASSIGN KJNF,JNF(1;3072)
                ASSIGN DJNA,JNA(1;49152)
                ASSIGN KJNS,JNS(1;1536)
                ASSIGN DJN1,JN1(1;26112)
                ASSIGN DC,JNA(1;17)
                CALL  Q3CLOCKS(IDATE,ITIME)
                WRITE(6,992) IDATE,ITIME
          992   FORMAT(' COMPILE VERSION 5-18-77, NEW FILT34',
             X       '.  DATE,TIME = ',2A12)
                JGAUSS=0
       C  SET SEED DEFAULT
                OPSEED = SDNORM
                JSEED = SDRUN
                IPRIN = 2
       C63      FORMAT(4F10.5,I5)
                READ(5,INSTR)
                WRITE(5,INSTR)
       C64      FORMAT(5F10.5,4I5)
       C
                IF(OPSEED .EQ. SDRDWR) GOTO  31
                      CALL   GAUSS(SDSTAR,   TEM1,TEM2,TEM3)
                      GOTO  32
           31   CONTINUE
                      CALL   GAUSS(SDREST,   TEM1,TEM2,TEM3)
           32   CONTINUE
       C
```

```
      P110=10.**(ALP110/10.)
      QQ=Q22C**(.25)
      RX=(P110/(SQRT(2.0)*QQ))**(4.0/3.0)
      FTC=SQRT(2.0)*RX**(.25)/QQ
      DELT=DELF*FTC
      Q22=Q22C*DELT
      R11=RX/DELT
      R11M1 = 1./R11
      P220=P110*SQRT(Q22C/RX)
      ALFD=ALF*DELT
      BET=1.0-ALFD
      A11=10.**((ALP110+GAM)/10.)
      A22=P220
      P330=0.5*Q33C/ALF
      A33=2.0*P330
      Q33=Q33C*DELT
      DEV1=SQRT(A11)
      DEV2=SQRT(A22)
      DEV3=SQRT(R11)
      DEV4=SQRT(A33)
      DEVQ2=SQRT(Q22)
      DEVQ3=SQRT(Q33)
      KOUNT=1
      ISAMP=1
      NSAMP=0
      SUMP1=0.0
      SUMP2=0.0
      SUMP3=0.0
      DELSQ=DELT**2
      PI=4.*ATAN(1.)
      PI2=2.0*PI
      PIDLT=PI/DELT
      PINV=1.0/PI
      PI2DLT=2.0*PIDLT
      U1=NUM1
      U2=NUM2
      U3=NUM3
      IY2=U2/PI2DLT*SQRT(50.0*Q22)+.5
      NTERM=IY2
      NTERM1=NTERM+1
      NTERM16=NTERM*16
      NC=NTERM16+1
      NT=2*NTERM16
      NTA1536=NT+1536
      R55=0.5/R11
```

```
          CL=-0.5/A22
          CM=-0.5/A33
          SI=-0.5/A11
C         ****************GRID Y1,Y2 AND YA ********************************
          EDG1=PI/U1
          EDG2=PIDLT/U2
          DO 40 I=1,NUM1
            X=I-1
            X=X/U1
   40       Y1(I)=-PI+X*PI2+EDG1
          DO 50 I=1,NUM2
            X=I-1
            X=X/U2
   50       Y2(I)=-PIDLT+X*PI2DLT+EDG2
          DO 51 I=1,IY2
            X=I
            X=X/U2
   51       Y3(I)=X
          DO 55 I=1,NUM1
            COSY(I)=COS(Y1(I))
   55       SINY(I)=SIN(Y1(I))
          S1(1;16)=COSY(1;16)/R11
          S2(1;16)=SINY(1;16)/R11
          CALL VPROP(SINY,1)
          CALL VPROP(COSY,1)
C        BEGIN NEW SAMPLE PATH
       71 XP=0.5*SQRT(A33)
          YY3=(NUM3-1.0)/2.0+1.0
          IY3=YY3
C
          IF(IPRIN .GE. 2)
         X    WRITE(6,981)
  981         FORMAT('1   KOUNT,X1,X2,X3,    Z1,Z2,    ',
         X        /' CXHAT,X3EST,ALOSS,  TNLF')
          DO 60 I=1,NUM3
            XX=I-YY3
            YC(I)=Y3EST+XP*XX
   60       YA(I)=Y3EST+XP*XX
C         ******************** DYNAMIC EXPONENTIALS ************************
          IF(IY2.EQ.0)GO TO 153
          DO 150 I=1,IY2
          DON=PIDLT*PI2DLT
  150     EXDON(I)=EXP(-DON/Q22*(Y3(I)**2))
          DO 152 I=1,IY2
            EXPON(I)=EXDON(IY2+1-I)
  152     EXPON(IY2+1+I)=EXDON(I)
  153     EXPON(IY2+1)=0.5
         IYY=2*IY2+1
         LTERM2=0
         DO 720 K=1,16
         DO 720 I=1,16
         XNUM=(K-I)*XP+ALFD*(YA(I)-1.)
         XNUM=-0.5/Q33*XNUM**2
         EXP33(I,K)=0.0
         IF(XNUM.LT.-27.)GO TO 720
         EXP33(I,K)=EXP(XNUM)
  720    CONTINUE
         LTERM=0
         LTERM16=LTERM*16
         LTERM1=LTERM+1
         LC=LTERM16+1
         NS=NTA1536*LTERM+NC
         I=0
         DO 339 K=1,16
         DO 339 N=1,16
         DO 339 J=NTERM1,IYY
         I=I+1
  339    D(I)=EXPON(J)*EXP33(N,K)
         ALOSS=A33
         ALOSSO=ALOSS
         X3EST = Y3EST
         X3ESTO = Y3EST
C        ****************************** INITIAL DENSITY ********************
         IJK=0
         DO 160 K=1,16
         ZZZ=CM*(YA(K)-Y3EST)**2
         DO 160 J=1,96
         YYY=ZZZ+CL*(Y2(J)-Y2EST)**2
         DO 160 I=1,16
         IJK=IJK+1
         XXX=YYY+SI*(Y1(I)-Y1EST)**2
         IF(XXX.LT.-27.)GO TO 159
         JN(IJK)=EXP(XXX)
         GO TO 160
  159    JN(IJK)=0.0
  160    CONTINUE
C        ****************** INTEGER ARRANGEMENT ***************************
         DO  225 I=1,26112
  225    B3(I)=B'1'
         I=0
```

```
              DO 300 K=1,16
              DO 300 J=1,96
              I=I+17
       300    B3(I)=B'0'
              DO 320 K=1,3071,2
              JNF(K)=(K-1)*8
       320    JNF(K+1)=JNF(K)
              I=0
              J1=0
              DO  332 K=1,16
              DO 332 J=1,96
              I=I+1
              I1=J1+MOD(23-(J-1)/6,16)
              JNS(I)=I1
       332    J1=J1+32
              DELJ(1;17)=11./12.
              DELJ(18;17)=0.75
              DELJ(35;17)=7./12.
              DELJ(52;17)=5./12.
              DELJ(69;17)=0.25
              DELJ(86;17)=1./12.
              CALL VPROP1(DELJ)
       11     CONTINUE
              KOUNT=1
              CALL GAUSS(JSEED,DEV1,Y1EST,X1)
              XDAT(KOUNT,1)=X1
              CALL GAUSS(JSEED,DEV2,Y2EST,X2)
              CALL GAUSS(JSEED,DEV4,Y3EST,X3)
              ACOS=X3*COS(X1)
              ASIN=X3*SIN(X1)
              CALL GAUSS(JSEED,DEV3,ACOS,Z1)
              CALL GAUSS(JSEED,DEV3,ASIN,Z2)
              GO TO 470
       450    CONTINUE
              X1=X1+X2*DELT
              XDAT(KOUNT,1)=X1
              CALL GAUSS(JSEED,DEVQ2,X2,X2)
              X3=X3*BET+ALFD
              CALL GAUSS(JSEED,DEVQ3,X3,X3)
              XDAT(KOUNT,5)=X3
              ACOS=X3*COS(X1)
              ASIN=X3*SIN(X1)
              CALL GAUSS(JSEED,DEV3,ACOS,Z1)
              CALL GAUSS(JSEED,DEV3,ASIN,Z2)
        C     XP=0.5*AMAX1(.001,SQRT(ALOSS))
```

```
            XP=.5*SQRT(ALOSS)
            XPD=.5*SQRT(ALOSSD)
C           XPD=.5*AMAX1(.001,SQRT(ALOSSD))
            DO 600 I=1,NUM3
              XX=I-YY3
              YA(I)=X3ESTO+XX*XPD
      600   CONTINUE
            DO 730 K=1,16
            DO 730 I=1,16
            XNUM=-YA(I)+X3EST+XP*(K-YY3)+ALFD*(YA(I)-1.)
            XNUM=-0.5/Q33*XNUM**2
            EXP33(I,K)=0.0
            IF(XNUM.LT.-27.)GO TO 730
            EXP33(I,K)=EXP(XNUM)
      730   CONTINUE
            I=0
            DO 340 K=1,16
            DO 340 N=1,16
            DO 340 J=NTERM1,IYY
            I=I+1
      340   D(I)=EXPON(J)*EXP33(N,K)
      470   CONTINUE
C           ************************* SENSOR FUNCTION *************************
            CALL Q3CLOCKS(T,TT)
            R11TZ1 = Z1*R11M1
             R11TZ2 = Z2*R11M1
             S1(1;16) = R11TZ1*COSY(1;16)
             S2(1;16) = R11TZ2*SINY(1;16)
            S1(1;16)=S1(1;16)+S2(1;16)
            J=1
            DO  500 K=1,16
            S2(1;16)=S1(1;16)*(X3EST+(K-YY3)*XP)
            SN2(1;16)=VEXP(S2(1;16);SN2(1;16))
            CALL VPROP(SN2,0)
            SN1(J;1536)=SN2(1;1536)
            S2(1;16)=-R55*(X3EST+(K-YY3)*XP)*(X3EST+(K-YY3)*XP)
            SN2(1;16)=VEXP(S2(1;16);SN2(1;16))
            CALL VPROP(SN2,0)
            SN1(J;1536)=SN1(J;1536)*SN2(1;1536)
      500   J=J+1536
C           ******************** MAIN LOOPOSTARTS ******************************
            JN(1;24576)=JN(1;24576)*SN1(1;24576)
            CALL Q8VXTOV(X'02',0,KJNF,0,DB,0,DJNA)
            CALL Q8VXTOV(X'02',0,KJNS,0,DC,0,DJN1)
            JNA(1;26112)=JN1(2;26112)-JN1(1;26112)
```

```
       JNA(1;26112)=DELJ(1;26112)*JNA(1;26112)
       JN1(1;26112)=JN1(1;26112)+JNA(1;26112)
       CALL  PRVEC('JN1', JN1)
       JNA(1;24576)=Q8VCMPRS(JN1(1;26112),B3(1;26112);JN1(1;24576))
       JN1(1;24576)=JNA(1;24576)
       J=1
       I=1
       DO 510 K=1,16
       N=I+NTERM16
       JNA(N;1536)=JN1(J;1536)
       JNA(I;NTERM16)=JNA(I+1536;NTERM16)
       JNA(N+1536;NTERM16)=JN1(J;NTERM16)
       J=J+1536
 510   I=I+NTA1536
       N=1
       I1=0
       JK=1
       CALL  PRVEC('JNA', JNA)
       DO 700 I=1,16
       I1=0
       JN1(N;1536)=0.0
       DO 690 K=1,16
       J1=NS+I1
       J2=NS+I1
       DO 680 J=1,NTERM1
       JN(1;1536)=JNA(J1;1536)+JNA(J2;1536)
       JN(1;1536)=JN(1;1536)*D(JK)
       JN1(N;1536)=JN1(N;1536)+JN(1;1536)
       JK=JK+1
       J1=J1+16
 680   J2=J2-16
 690   I1=I1+NTA1536
       N=N+1536
 700   CONTINUE
       CALL PRVEC('JN1T',JN1)
       CNORM=SUMLOG(JN1)
       IF(CNORM.LT.1.0E-20)CNORM=1.0
       CNORM=1./CNORM
       JN(1;24576)=CNORM*JN1(1;24576)
       SHAT = 0.
       CHAT = 0.
       SUMSC = 0.
 C   3-2-77
       DO 751 I=1,16
          SUMSC = 0.
```

```
                DO 729 J=1,96
                DO 729 K=1,16

                   ITEMP = I+16*(J-1)+1536*(K-1)
                   SUMSC = SUMSC+JN(ITEMP)
         729    CONTINUE
                CHAT = CHAT+SUMSC*COSY(I)
                SHAT = SHAT+SUMSC*SINY(I)
         751  CONTINUE
      C       JNA(1;24576)=COSY(1;24576)*JN(1;24576)
      C       CHAT=SUMLOG(JNA)
      C       JNA(1;24576)=SINY(1;24576)*JN(1;24576)
      C       SHAT=SUMLOG(JNA)
              CXHAT=ATAN2(SHAT,CHAT)
              J=1
              DO 343 K=1,16
              YB(J;1536)=(X3EST+(K-YY3)*XP)
         343  J=J+1536
              JNA(1;24576)=YB(1;24576)*JN(1;24576)
              X3ESTO=X3EST
              X3EST=SUMLOG(JNA)
              JNA(1;24576)=JNA(1;24576)*YB(1;24576)
              ALOSSO=ALOSS
              ALOSS=SUMLOG(JNA)
              ALOSS=AMAX1(ALOSS-X3EST*X3EST,1.E-18)
      C       *************************** MAIN LOOP ENDS ***********************
              CALL Q3CLOCKS(TNLF,TT)
              IF(IPRIN .GE. 2)
             X     WRITE(6,201)KOUNT,X1,X2,X3,Z1,Z2,CXHAT,X3EST,ALOSS
         201  FORMAT(1H ,   I5,1X,1P3E14.6,4X,1P2E14.6,4X,1P3E14.6 )
              IF(IPRIN .GE. 2)
             X     WRITE(6,8880)TNLF
        8880  FORMAT(1H ,1PE12.6)
              IF(KOUNT.EQ.NO2)GO TO 505
              XDAT(KOUNT,2)=CXHAT
              XDAT(KOUNT,3)=ALOSS
              XDAT(KOUNT,4)=X3EST
              KOUNT=KOUNT+1
              GO TO 450
         505  CONTINUE
              SUMP=0.0
              SUMC=0.0
              XDAT(KOUNT,2)=CXHAT
              XDAT(KOUNT,3)=ALOSS
              XDAT(KOUNT,4)=X3EST
      DO 1501 I=31,NO2
         XD=ABS(XDAT(I,1)-XDAT(I,2))
 1498 CONTINUE
         IF(XD.GT.PI)GO TO 1499
      GO TO 1500
 1499 XD=XD-PI2
      GO TO 1498
 1500 SUMP=SUMP+XD**2
 1501 CONTINUE
      H=NO2-30
      SUMP=SUMP/H
      XNSAMP=NSAMP
      XAA=XNSAMP+1.0
      SUMP1=(SUMP+XNSAMP*SUMP1)/XAA
      DSUMP1=ALOG10(SUMP1)*10.0
      DO 1601 I=31,NO2
         XD=ABS(XDAT(I,5)-XDAT(I,4))
 1698 CONTINUE
         IF(XD.GT.PI)GO TO 1699
      GO TO 1700
 1699 XD=XD-PI2
      GO TO 1698
 1700 SUMC=SUMC+XD**2
 1601 CONTINUE
      SUMC=SUMC/H
      SUMP2=(SUMC+XNSAMP*SUMP2)/XAA
      DSUMP2=ALOG10(SUMP2)*10.0
      WRITE(6,1511)NSAMP,SUMP1,DSUMP1,SUMP2,DSUMP2
 1511 FORMAT(1H ,I10,1P4E14.6)
      NSAMP=NSAMP+1
C
C   OPTIONAL SAVE OF SEED
      IF( (OPSEED .EQ. SDWRT) .OR. (OPSEED .EQ. SDRDWR))
     X   CALL GAUSS(SDSAVE,   TEM1,TEM2,TEM3)
C
      IF(ISAMP .EQ. NO3) GO TO 2200
      ISAMP = ISAMP+1
      GO TO 71
 2200 CONTINUE
C
      STOP
      END
```

```
FORTRAN R1.3 CYCLE I         BUILT 09/27/78 20:40      SOURCE LISTING
      FUNCTION SUMLOG(A)
      REAL A(26112),C(12288)
      C(1;12288)=A(1;12288)+A(12289;12288)
      C(1;6144)=C(1;6144)+C(6145;6144)
      C(1;3072)=C(1;3072)+C(3073;3072)
      C(1;1536)=C(1;1536)+C(1537;1536)
      C(1;768)=C(1;768)+C(769;768)
      C(1;384)=C(1;384)+C(385;384)
      C(1;192)=C(1;192)+C(193;192)
      C(1;96)=C(1;96)+C(97;96)
      C(1;48)=C(1;48)+C(49;48)
      C(1;24)=C(1;24)+C(25;24)
      C(1;12)=C(1;12)+C(13;12)
      C(1;6)=C(1;6)+C(7;6)
      C(1;3)=C(1;3)+C(4;3)
      SUMLOG=C(1)+C(2)+C(3)
      RETURN
      END
```

```
FORTRAN R1.3 CYCLE I          BUILT 09/27/78 20:40     SOURCE LISTING
                   SUBROUTINE VPROP(A,I)
                   REAL A(24576)
                   IF(I.EQ.2)GO TO 10
                   A(17;16)=A(1;16)
                   A(33;32)=A(1;32)
                   A(65;32)=A(1;32)
              10   A(97;96)=A(1;96)
                   A(193;192)=A(1;192)
                   A(385;384)=A(1;384)
                   A(769;768)=A(1;768)
                   IF(I.EQ.0)RETURN
                   A(1537;1536)=A(1;1536)
                   A(3073;3072)=A(1;3072)
                   A(6145;6144)=A(1;6144)
                   A(12289;12288)=A(1;12288)
                   RETURN
                   END
```

```
FORTRAN R1.3 CYCLE I          BUILT 09/27/78 20:40      SOURCE LISTING
                  SUBROUTINE VPROP1(A)
                  REAL A(26112)
                  A(103;102)=A(1;102)
                  A(205;204)=A(1;204)
                  A(409;408)=A(1;408)
                  A(817;816)=A(1;816)
                  A(1633;1632)=A(1;1632)
                  A(3265;3264)=A(1;3264)
                  A(6529;6528)=A(1;6528)
                  A(13057;13056)=A(1;13056)
                  RETURN
                  END
```

```
FORTRAN R1.3 CYCLE I          BUILT 09/27/78 20:40     SOURCE LISTING
      FUNCTION RNNF(NS,MODE)
      DIMENSION  NS(2), NC(2)
      COMMON /RN/ N1, N2, MP, T1, T2
      DATA  M1, M2/244734, 158551/
C         MODE=0 TO CONTINUE, OTHERWISE RESTART WITH
C         INTEGER NUMBER NS(1)*2**18+NS(2)
      IF (MODE) 10, 100, 10
   10 N1=NS(1)
      N2=NS(2)
      T1=2.**(-18)
      T2=2.**(-36)
      MP=2**18
      RETURN
C
  100 DO 200 I=1,2
      GO TO (110,120),I
  110 K=M2*N2
      GO TO 190
  120 K=M1*N2+M2*N1+KD
  190 KD=K/MP
  200 NC(I)=K-KD*MP
      N1=NC(2)
      N2=NC(1)
      XN1=N1
      XN2=N2
      RNNF=XN1*T1+XN2*T2
      RETURN
      END
```

```
FORTRAN R1.3 CYCLE I          BUILT 09/27/78 20:40     SOURCE LISTING
      SUBROUTINE GAUSS(JS,SD,XM,X)
      DIMENSION  NST(2)
      COMMON /RN/ N1, N2, MC, T1, T2
      COMMON /GN/ J,XR(2)
C   SELECT  RESTART,RUN,SAVE,RESTORE
C
      GOTO (10, 20, 101, 201),JS
   10  J=1
      TWOPI=8.*ATAN(1.)
      NST(1)=244734
      NST(2)=158551
      NST(1)=102943
      NST(2)=185617
      XR(1)=RNNF(NST,1)
      RETURN
C
C     RUN (GENERATE RANDOM NO.)
   20 GO TO (30,40), J
   30 J=2
      XR(1)=RNNF(NST,0)
   35 XR(2)=RNNF(NST,C)
      X1=SQRT(ABS(-2.*ALOG(XR(1))))
      XR(2)=TWOPI*XR(2)
      XR(1)=X1*SIN(XR(2))
      XR(2)=X1*COS(XR(2))
      X=XR(1)*SD+XM
      RETURN
   40 J=1
      X=XR(2)*SD+XM
      RETURN
C
C    SAVE SEED
  101  REWIND 10
      WRITE(10,991) N1,N2,J,XR(2)
      WRITE(6,991) N1,N2,J,XR(2)
      RETURN
C
C     RESTORE SEED
 201  CONTINUE
      REWIND 10
      READ(10,991) NST(1),NST(2),J,XR(2)
       WRITE(6,991) NST(1), NST(2), J,XR(2)
      TWOPI = 8.*ATAN(1.)
      XR(1) = PNNF(NST,1)
      RETURN
```

```
FORTRAN R1.3 CYCLE I          BUILT 09/27/78 20:40     SOURCE LISTING           GAUSS
                C
                991 FORMAT(' RANDOM SEEDS ',3I10,E28.18)
                    END
```

```
FORTRAN R1.3 CYCLE I        BUILT 09/27/78 20:40     SOURCE LISTING
                  SUBROUTINE  PRVEC(LABEL,VEC)
          C    PRINT  SELECTED VECTOR COMPONENTS
                  INTEGER  LABEL,DIM12,COUNT
                  REAL VEC(1)
                  DATA  DIM12,COUNT/1536, 0/
                  IF(COUNT .GE. 0) RETURN
                  COUNT = COUNT+1
                  WRITE(6,99) COUNT, LABEL,
                *  VEC(DIM12*7+1), VEC(DIM12*7+760),VEC(DIM12*7+1148)
             99   FORMAT(' PR. ENTRY ',I5,' AT PNT. ',A4,
                *   5E14.7)
                  RETURN
                  END
```

II - 3 The AP120B Fortran, Assembly Language, Vector Chainer

By referring to figure 1 the structure of the code for the Monte Carlo restartable code for the 3D phase demodulation becomes clear. The code is made up of three different types, Fortran Code, Assembly Language AP120B Code, called as a Fortran Subroutine Vector Chained AP120B Code, which is the concatenation of Assembly Language Codes. The restartable features of the code, the current Monte Carlo averages are written to a file after each sample path, were provided by Milt Campbell. This program was used to generate the statistical data provided in [2] . The time critical convolution loops are realized by the assembly codes RLNLF.FSO and STHIRD.FSO which convolve over phase rate and amplitude respectively. The coding of these loops are time optimal for the AP120B.

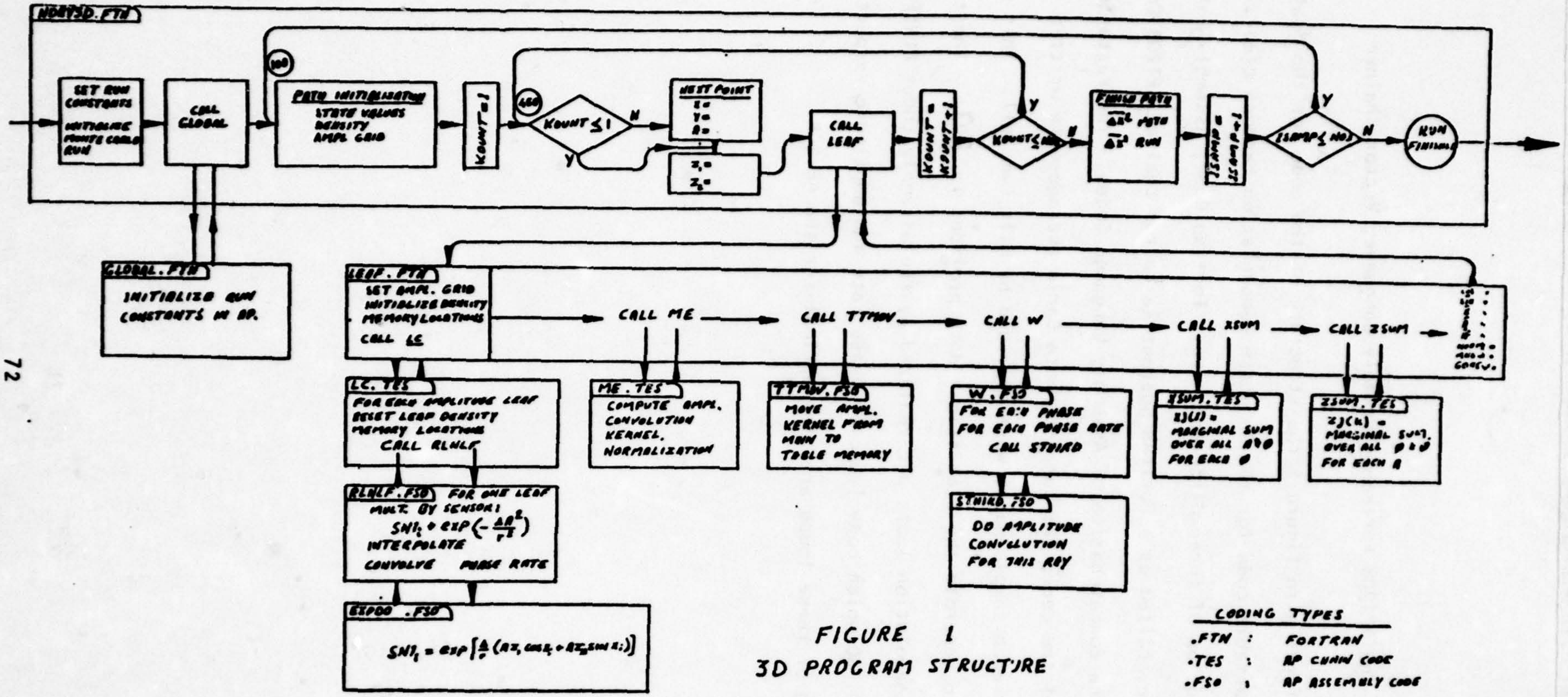


FIGURE 1
3D PROGRAM STRUCTURE

CODING TYPES

.FTN	:	FORTRAN
.TES	:	AP CHAIN CODE
.FSO	:	AP ASSEMBLY CODE

THIS FILE CONTAINS INFORMATION ON HOW TO RUN EXGO.

M. CAMPBELL (SYSCON DESIGN)
OCTOBER 9,1978

SECTIONS ARE PRECEDED BY A LINE
##N
WHERE N IS THE NUMBER. THIS ALLOWS EASY ACCESS VIA THE EDITOR.

CONTENTS.

##1 RUNNING THE PROGRAM

##1.1 THE PROGRAM IS ON FILE EXGO.TSK SO IT MAY BE RUN AS A NORMAL RSX-11M PROGRAM. ON STARTUP THE PROGRAM EXPECTS THE FILES 'INITIAL.DAT' AND 'RESTART.DAT' TO BE PRESENT AND TO CONTAIN THEIR CORRECT VALUES (SEE DATA FILE STRUCTURE). 'INITIAL.DAT' CONTAINS DATA CONSTANT FOR A RUN AND ONCE EDITTED TO YOUR SATISFACTION NEED NOT BE MODIFIED. 'RESTART.DAT' IS DYNAMICALLY UPDATED BY THE PROGRAM AND CONTAINS THE CURRENT RESTART INFORMATION. FOR THE INITIAL RUN OF THE PROGRAM ONLY!, FILE 'RESTART.INT' SHOULD BE COPIED TO 'RESTART.DAT' TO ENSURE THAT A RUN OF THE PROGRAM WILL INITIALIZE PROPERLY.

##1.2 COMMAND FILE 'NEWRUN.DAT' IS PROVIDED TO SET UP THE DATA FILES FOR INITIATING AN EXGO RUN. IT RENAMES ANY OLD RESTART FILES (WHICH CONTAIN FINAL RESULTS OF RUNS) TO BE 'RESTART.OLD', DELETES ANY EXISTING BACKUP FILES, CREATES AN INITIAL RESTART BY COPYING 'RESTART.INT' TO 'RESTART.DAT' AND EXITS. THE FILES ARE NOW READY FOR AN 'RUN EXGO' COMMAND.

##1.3 PROGRAM CTLXGO IS PROVIDED TO ALLOW ORDERLY SHUT DOWN OF EXGO EXTERNALLY. EXGO USES EVENT FLAG 54 FOR CONTROL. IF, AT THE END OF THE MAIN LOOP, THE EVENT FLAG IS SET, EXGO SHUTS DOWN WITH THE DATA FILES SET UP FOR RESTART.

##2. GETTING DATA

##2.1 RUNNING VALUES. THE FILE 'RESTART.DAT' ALWAYS CONTAINS THE RESULTS OF THE LAST TIME THROUGH THE OUTER LOOP OF EXGO. IT IS THIS FILE THAT WILL BE USED IF EXGO IS INTERRUPTED AND THEN RESTARTED. EXAMINING 'RESTART.DAT' OULD PROVIDE THE LATEST INFORMATION ON THE STATUS OF EXGO.

THE FILE 'BACKUP.DAT' CONTAINS THE SAME VALUES AS 'RESTART.DAT' BUT FROM THE PREVIOUS PASS THROUGH THE OUTER LOOP. THIS IS THE SECONDARY RECOVERY FILE IN CASE THERE IS SOME PROBLEM WITH 'RESTART.DAT'.

##2.2 START STATUS. A NEW VERSION OF 'RUNSTAT.DAT' IS CREATED EACH TIME EXGO IS STARTED AND ANYTIME THE PROGRAM FAILS

##2.3 EXAMINING THE FILES. 'RESTART.DAT' AND 'BACKUP.DAT' SHOULD BE EXAMINED ONLY WITH EXGO NOT RUNNING, SINCE EXGO WILL QUIT (WITH AN ERROR ON 'RUNSTAT.DAT') IF IT CAN NOT ACCESS BOTH FILES.

'RUNSTAT.DAT' MAY BE EXAMINED AT ANY TIME AS A NEW VERSION IS CREATED AS NEEDED.

##3. RECOVERING FROM ERRORS.

##3.1 IF EXGO ATTEMPTS TO KEEP THE RESULTS OF THE LAST TIME THROUGH THE OUTER LOOP ON THE FILE 'RESTART.DAT' AND THE RESULTS ON THE PREVIOUS PASS ON 'BACKUP.DAT'. THE OLD DATA IS COPIED FROM 'RESTART.DAT' TO 'BACKUP.DAT' BEFORE WRITING THE NEW DATA TO 'RESTART.DAT'.

IF EXGO IS UNABLE TO ACCESS ANYONE OF 'INITIAL.DAT', 'RESTART.DAT' OR 'BACKUP.DAT', OR IF THERE IS SOME ERROR IN READING THEM (END-OF-FILE OR CONSISTENCY CHECK BAD), IT WRITES AN ERROR MESSAGE ON 'RUNSTAT.DAT' AND STOPS.

##3.1 IF 'RESTART.DAT' IS BAD BUT 'BACKUP.DAT' IS GOOD, RENAME 'BACKUP.DAT' TO BE 'RESTART.DAT'. THIS RESULTS IN THE LOSS OF ONE PASS THROUGH THE OUTER LOOP.

##3.2 IF BOTH 'RESTART.DAT' AND 'BACKUP.DAT' ARE BAD, THE LATEST 'RUNSTAT.DAT' CAN BE USED BY RENAMING IT TO BE 'RESTART.DAT' AND EDITTING THE TIME TAG (LINE 1) OUT. THIS RESULTS IN LOSS OF ALL DATA SINCE THE LAST SUCCESSFUL RESTART.

THIS SHOULD BE A VERY RARE CASE SINCE EXGO DOES NOT HAVE MORE THAN ONE OF ANY OF ITS FILES OPEN AT ONCE.

##4. DATA FILE FORMAT.

##4.1 RESTART.DAT-THE MAIN RECOVERY DATA FILE. IT IS ACCESSED EACH TIME EXGO IS STARTED FOR THE RUNNING VALUES TO BE USED.

FORMAT: (NOTE, THE ACTUAL FILE HAS COMMA'S AFTER SOME VALUES, THESE ARE FOR EASE IN EDITING AND SHOULD BE RETAINED)

LINE	USE
1	CURRENT VALUE FOR ISAMP
2	CURRENT VALUE FOR NSAMP
3	CURRENT VALUE FOR SUMP1
4	CURRENT VALUE FOR SUMP2
5	CURRENT VALUE FOR JGAUSS
6	CURRENT VALUE FOR DZZZ1
7	CURRENT VALUE FOR XZZZ(1)
8	CURRENT VALUE FOR XZZZ(2)
9	COMMENT LINE(NO DATA IS ON THIS LINE)
10	6 INTERNAL VALUES FOR GAUSS (THE ARRAY NST)
11	7 INTERNAL VALUES FOR BANF (N1 TO N6 AND NP)
12	3 INTERNAL VALUES FOR BANF (T1 TO T3)
13	3 INTERNAL VALUES FOR BANF (T4 TO T6)
14	THE INTEGER VALUE '12345' IS REQUIRED. EXGO USES THIS AS A CHECK TO MAKE SURE THE FILE WAS CORRECTLY WRITTEN.

##4.2 INITIAL.DAT-CONTAINS CONSTANT DATA FOR A RUN, BUT THAT MAY VARY BETWEEN RUNS. THIS FILE MAY BE EDITED TO CHANGE RUN CHARACTERISTICS.

FORMAT:

LINE	USE
1	IPRNT-IF NON-ZERO THEN THE 'CYCLIC INPUT' DATA IS LISTED
2	JPRNT-INNER LOOP DATA (IN NDRV3D AND LEAF) IS LISTED WHEN MOD(KOUNT,JPRNT) IS ZERO. SET IT LARGER THAN NO2 IF NO DATA IS DESIRED.
3	KPRNT-RUNNING RESULTS ARE PRINTED WHEN MOD(ISAMP,KPRNT) IS ZERO. SET TO PRINT INTERVAL DESIRED.
4	ALP110
5	DELF
6	Q22C
7	Q33C
8	NO2
9	NO3
10	ALF

##4.3 RUNSTAT.DAT-A NEW VERSION OF THIS FILE IS CREATED EACH TIME IT IS NEEDED. IT EITHER CONTAINS THE TIME AND DATE OF A SUCCESSFUL RESTART, WITH THE RESTART DATA IN 'RESTART.DAT' FORMAT OR AN ERROR MESSAGE.

##4.4 RESTART.INT-THIS FILE CONTAINS THE INITIAL VALUES OF 'RESTART.DAT', SO A NEW RUN WILL INITIALIZE PROPERLY. IT HAS THE SAME FORMAT AS 'RESTART.DAT'.

##4.5 BACKUP.DAT-THIS FILE IS A COPY OF 'RESTART.DAT' MADE BEFORE WRITING NEW VALUES TO THE RESART FILE. IT HAS THE SAME FORMAT AS 'RESART.DAT'.

##5. FORTRAN SOURCE FILES

##5.1 NDRV3D.FTN

THIS FILE CONTAINS THE SAME ROUTINES AS IT ORIGINALLY DID, HOWEVER NDRV3D ITSELF (THE MAIN PROGRAM) HAS BEEN HEAVILY MODIFIED TO INSTALL THE RESTART CAPABILITY. MINOR MODS TO GAUSS AND BANF TO INCLUDE THEIR REMEMBERED VALUES IN COMMON SO THEY CAN BE WRITTEN TO FILES.

##5.2 LEAF.FTN

THIS FILE CONTAINS THE SAME ROUTINES AS IT ORIGINNALLY DID, LEAF HAS BEEN SLIGHTLY MODIFIED TO MAKE THE PRINTING OF DATA AT THE END OF EACH CALL OPTIONAL.

##5.3 KILLME.FTN

SUBROUTINE KILLME IS CALLED AFTER SETTING UP THE RECOVERY FILES TO SEE IF EVENT FLAG 54 HAS BEEN SET. IT SO IT EXECUTES A STOP.

##5.4 ERROR.FTN

SUBROUTINE ERROR IS CALLED WHEN EXGO DISCOVERS ANY ERROR DURING A RESTART ATTEMPT OR WHEN TRYING TO SET UP THE RESTART FILES. ERROR CREATES A VERSION OF 'RUNSTAT.DAT'

```
        CONTAINING AND ERROR MESSAGE AND STOPS.

##5.5   CTLXGO.FTN
        PROGRAM CTLXGO IS AN INDEPENDENT PROGRAM THAT SETS EVENT
        FLAG 54 SO THAT EXGO WILL STOP ON ITS NEXT PASS THROUGH
        THE OUTER LOOP.

##6.  COMMAND FILES

##6.1  TEST.CMD
        CONTAINS THE NECESSARY COMMANDS TO TKB EXGO.

##6.2  NEWRUN.CMD
        CONTAINS THE NECESSARY COMMANDS TO REINIALIZE THE
        DATA FILES FOR A COMPLETELY NEW RUN OF EXGO.


>
```

```
.ENABLE DATA
.OPEN BOXBLD.CMD
EXGO/FP/CP,EXGO/CR/-WI=BOXBLD/MP
UNITS=10
ASG=AP:8,AP1:9,AP2:10
PRI=10
//
.CLOSE
.OPEN BOXBLD.ODL
        .ROOT MAIN-*(KILL,ERR,GLOB,REST)
MAIN:   .FCTR NDRV3D-[1,1]FPSLIB/LB:APINIT-[1,1]FPSLIB/LB-[1,1]SHORT
ERR:    .FCTR ERROR
KILL:   .FCTR KILLME
GLOB:   .FCTR [340,340]GLOBAL-[1,1]FPSLIB/LB
REST:   .FCTR REST1-REST2-REST3-REST4-[1,1]FPSLIB/LB
REST1:  .FCTR LEAF-[340,340]LC-[1,1]FPSLIB/LB
REST2:  .FCTR [340,340]M-[340,340]W-[1,1]FPSLIB/LB
REST3:  .FCTR [340,340]ZSUM-[340,340]XSUM-[1,1]FPSLIB/LB
REST4:  .FCTR [340,340]TTMOV-[1,1]FPSLIB/LB
        .END
.CLOSE
PIP EXGO.TSK;*/DE
TKB @BOXBLD
PUR EXGO.*,BOXBLD.*

>
```

```
      C** NDRV3D.FTN
      C     NDRV3D: NEW 3D DRIVER LINEAR LOGGIC
      C        VERSION 5/28/78
      C MODIFIED FOR AUTO RESTART 10/4/78 (M.CAMPBELL)
      C
               REAL J0(1536),J00(1536),XDAT(130,10),NORM,MNEW,MOLD
               INTEGER SN1Z,SINFZ,COSFZ,DELZ,AZ,S1Z,S2Z,T1Z,T2Z
               INTEGER H
               INTEGER COSF,SINF,CEIL,AGOOLD,AMOLD,AGONEW,AMNEW
               INTEGER AAOLD,AANEW,ASC1,ASC2,ASC3,AA2R,AXP1,AADLT
               INTEGER ASS
               INTEGER AXP2,AGA,ACLF,AAM1,AAM2,AZJ,AXJ,ANORM,ASJ

               BYTE MYDATE(9),MYTIME(8)
      C THIS COMMON BLOCK CONTAINS PRINT CONTROL VARIABLE
               COMMON/PRINTC/IPRNT,JPRNT,KPRNT,KOUNT
               COMMON M,N,KMAX,A11,A22,Q33C,PIDLT,ALF,DELT,CONST,R11,
             1 MNEW,MOLD,GONEW,GOOLD,PI,TWOPI,Y1EST,Y2EST,Y3EST,
             2 CHAT,SHAT,XHAT,NORM,J0,Z1,Z2,
             3 COSY(16),SINY(16),AM1
            COMMON /GN/ DZZZ1, JGAUSS, XZZZ(2)
      C THIS COMMON CONTAINS GAUSS INTERNAL VARIABLE FOR RESTART
            COMMON/GSEED/INTRNL(6)
      C THIS COMMON CONTAINS BANF INTERNAL VARIABLES FOR RESTART
               COMMON/BFINT/IBNF(7),TBNF(6)
               COMMON INFLAG,LCHAT,LSHAT,SN1Z,COSFZ,SINFZ,DELZ,JNSZ,JZZ,
             1 MEMS,AZ,S1Z,S2Z,INBUFZ,T1Z,T2Z,ITOPS,ALDLT,GA,Q33,COSF,
             2 SINF,KBIAS,CEIL,AGOOLD,AMOLD,AGONEW,AMNEW,AAOLD,AANEW,ASC1,
             3 ASC2,ASC3,AA2R,AXP1,AADLT,AGA,AXP2,ACLF,AAM1,AAM2,AZJ,AXJ,
             4 ANORM,ASJ,ASS
      C
      C *************************  START RUN INITIALIZATION  ***************
               NORM=1.0
      C MYFLAG IS THE EVENT FLAG USED TO CONTROL EXGO
               MYFLAG=54
               CALL CLREF(MYFLAG)
               CALL DATE(MYDATE)
               CALL TIME(MYTIME)
      C
               M=16
               N=96
               KMAX=16
            IDEV=5
      C     BOX MEMORY ALLOCATIONS
            INFLAG=17
            LCHAT=18
            LSHAT=19
            SN1Z=20
            COSFZ=SN1Z+M
            SINFZ=COSFZ+M
            DELZ=SINFZ+M
            JNSZ=DELZ+N
            JZZ=JNSZ+N
            MEMS=JZZ+M*N+M
            AZ=MEMS+11
            S1Z=AZ+11
      S2Z=S1Z+M
      INBUFZ=S2Z+M
      T1Z=INBUFZ+2
      T2Z=T1Z+M
      ITOPS=AZ+21+4*M
      ALF=1.
C READ GENERAL PARAMETERS FROM FILE
        OPEN(UNIT=1,NAME='INITIAL.DAT',TYPE='OLD',ERR=5000)
        READ(1,9999,END=5000)IPRNT,JPRNT,KPRNT,ALP110,DELF,Q22C,Q33C,
     X  NO2,NO3,ALF
9999    FORMAT(3(I5,/),4(E15.8,/),2(I5,/),E15.8)
        CLOSE(UNIT=1)
      IF(IPRNT.NE.0)WRITE(IDEV,651) Y1EST,Y2EST,ALP110,DELF,Q22C,NO2
  651 FORMAT(' ',' CYCLIC INPUT'/4X,5F10.5,1I5)
      P110=10.**(ALP110/10.)
      QQ=Q22C**(.25)
      RX=(P110/(SQRT(2.0)*QQ))**(4.0/3.0)
      FTC=SQRT(2.0)* RX**(.25) /QQ
      DELT=DELF*FTC
      Q22=Q22C*DELT
        Q33=Q33C*DELT
      R11=RX/DELT
      P220=P110*SQRT(Q22C/RX)
      ALDLT=ALF*DELT
      GA=1.-ALDLT
      A11=10.**((ALP110+1.4)/10.)
      A22=    P220
      P330=.5*Q33C/ALF
      A33=2.0*P330
      PI=3.1415926536
        PI2=2*PI
      TWOPI=2.0*PI
      PIDLT=PI/DELT
      CONST=-2.0*PIDLT*PIDLT/Q22
      DEV1= SQRT(A11)
      DEV2= SQRT(A22)
      DEV3= SQRT(R11)
      DEV4=SQRT(A33)
      DEVQ2=SQRT(Q22)
      DEVQ3=SQRT(Q33)
        Y1EST=0.
        Y2EST=0.
        Y3EST=1.
        IY2=96./2./PIDLT*SQRT(50.*Q22)+.5
        KOUNT=1
        COSF=20+M
        SINF=COSF+M
        KBIAS=M*(N+1)
        CEIL=ITOPS+KMAX*KBIAS
        AGOOLD=CEIL+1
        AMOLD=AGOOLD+1
        AGONEW=AMOLD+1
        AMNEW=AGONEW+1
        AAOLD=AMNEW+1
        AANEW=AAOLD+KMAX
        ASC1=AANEW+KMAX
        ASC2=ASC1+KMAX
        ASC3=ASC2+KMAX
        AA2R=ASC3+KMAX
        AXP1=AA2R+KMAX
        AADLT=AXP1+1
        AGA=AADLT+1
        AXP2=AGA+1
        ACLF=AXP2+1
        AAM1=ACLF+KMAX*KMAX
        AAM2=AAM1+1
        AZJ=AAM2+1
        AXJ=AZJ+KMAX
        ANORM=AXJ+M
        ASJ=ANORM+1
        ASS=ASJ+KMAX
C--------READ RESTART FILE
C NOTE-FIRST RUN IS CONTROLLED BY RESTART FILE VAUES ALSO
C
        OPEN(UNIT=1,NAME='RESTART.DAT',TYPE='OLD',ERR=5010)
        READ(1,9998,END=5015)ISAMP,NSAMP,SUMP1,SUMP2,JGAUSS,DZZZ1
     X  ,XZZZ,INTRNL,IBNF,TBNF,MYRSTR
C THIS FORMAT ALSO USED BY RECOVERY SETUP CODE AT END OF OUTER LOOP
9998    FORMAT(2(I15,/),2(E15.8,/),I15,/,3(E15.8,/),/,6I10,/
     X  ,7I10,/,2(3F15.5,/),I15)
        CLOSE(UNIT=1)
        IF(MYRSTR.NE.12345)GO TO 5020
C RESTART SUCCESSFULL
        GO TO 6000
C UNSUCCESSFUL RESTART BRANCHES
C
C UNABLE TO OPEN OR ACCESS CONSTANT FILE
C
5000    CONTINUE
        CALL ERROR(1,1)
C UNABLE TO OPEN PRIMARY RESTART FILE
5010    CONTINUE
        CALL ERROR(1,2)
C END-OF-FILE ON PRIMARY RESTART FILE
5015    CONTINUE
        CALL ERROR(1,3)
C CONSISTENCY VARIABE -MYRSTR- DOES NOT HAVE VALUE OF '12345'
5020    CONTINUE
        CALL ERROR(1,4)
C
C SUCCESSFUL RESTART
C
6000    CONTINUE
        OPEN(UNIT=1,NAME='RUNSTAT.DAT',TYPE='NEW')
        WRITE(1,9991)MYTIME,MYDATE
9991    FORMAT(1X,8A1,1X,9A1,' RESTART SUCCESSFUL')
        WRITE(1,9990)ISAMP,NSAMP,SUMP1,SUMP2,JGAUSS,DZZZ1,
     X  XZZZ,INTRNL,IBNF,TBNF
        CLOSE(UNIT=1)
C THIS FORMAT ALSO USED BY RECOVER SET UP CODE AT END OF OUTER LOOP
9990    FORMAT(I15,' ,ISAMP',/
     X  I15,' ,NSAMP',/
     X  E15.8,' ,SUMP1',/
     X  E15.8,' ,SUMP2',/
     X  I15,' ,JGAUSS',/
     X  E15.8,' ,DZZZ1',/
     X  E15.8,' ,XZZZ(1)',/
     X  E15.8,' ,XZZZ(2)',/
     X  ' THE FOLLOWING ARE INTERNAL TO GAUSS AND BANF',/
     X ,6I10,/,
     X 7I10,/
     X ,3E15.8,/
     X ,3E15.8,/
     X  '             12345 , FILE CONSISTENCY CHECK VALUE')
C
C************************* END RUN CONSTANTS ******************
C
        CALL GLOBAL
C
C ********************** START PATH INITIALIZATION  ***************
C
  100   CONTINUE
      CALL GAUSS(JSEED,DEV1,Y1EST,X1)
      KOUNT=1
      XDAT(KOUNT,1)=X1
      CALL GAUSS(JSEED,DEV2,Y2EST,X2)
      CALL GAUSS(JSEED,DEV4,Y3EST,X3)
        XDAT(KOUNT,5)=X3
        ACOS=EXP(X3-1.)*COS(X1)
        ASIN=EXP(X3-1.)*SIN(X1)
        DO 11 K=1,KMAX
        YY3=.5*(FLOAT(KMAX)+1.)
        G=.5*(FLOAT(K)-YY3)
        G=G*G*.5
        AMFAK=0.
        IF (G.GT.27.) GO TO 12
        AMFAK=EXP(-G )
   12   CONTINUE
        MN=M*N
        DO 10 I= 1,M
        DO 10 J=1,N
        L1=I+M*(J-1)
        L2=M*(N+1)*(K-1)+ITOPS
   10   JOO(L1)=JO(L1)*AMFAK
        L4=776
        CALL APPUT(JOO,L2,MN,2)
      CALL APWD
   11 CONTINUE
        GOOLD=1.0+(-.5-FLOAT(KMAX)/2.)*.5*SQRT(A33)
        MOLD=SQRT(A33)/2.
        GONEW=GOOLD
        MNEW=MOLD
        CALL APPUT(GOOLD,AGOOLD,1,2)
        CALL APPUT(MOLD,AMOLD,1,2)
        CALL APPUT(GONEW,AGONEW,1,2)
        CALL APPUT(MNEW,AMNEW,1,2)
        CALL APWD
  205 FORMAT('0',8X,'POSIT.',5X,'POSIT. MOD 2 PI',2X,'EST. POSIT.',9X
     */´Z1 AND Z2´,19X,´CYCLIC LOSS´,5X,´ K-B EST. AND P11´)
C
C ********************************* END PATH INITIALIZATION *************
C
C ********************** START POINTS ***********************************
C
  450 CONTINUE
        IF (KOUNT.LE.1) GO TO 5
        ACOS=EXP(X3-1.)*COS(X1)
        ASIN=EXP(X3-1.)*SIN(X1)
    5 CALL GAUSS(JSEED,DEV3,ACOS,Z1)
      CALL GAUSS(JSEED,DEV3,ASIN,Z2)
      X1=X1 + X2*DELT
      XDAT(KOUNT,1)=X1
      CALL GAUSS(JSEED,DEVQ2,X2,X2)
        X3=GA*X3+ALDLT
      CALL GAUSS(JSEED,DEVQ3,X3,X3)
        XDAT(KOUNT,5)=X3
        CALL LEAF
        XDAT(KOUNT,2)=XHAT
        XDAT(KOUNT,4)=AM1

      IF(MOD(KOUNT,JPRNT).EQ.0)WRITE(IDEV,201)KOUNT,XDAT(KOUNT,1),
     X XDAT(KOUNT,2),Z1,Z2,
     *(XDAT(KOUNT,5)),AM1
  201 FORMAT(´O´,I3,1X,1P2E14.6/4X,1P2E14.6,4X,1P2E14.6 /)
      KOUNT=KOUNT + 1
        IF (KOUNT.LE.NO2) GO TO 450
C
C ************************ END POINTS ********************
C
C ********************** START FINISH PATH ***************
C
      SUMP=0.0
      SUMC=0.0
      DO 1501 I=31,NO2
        XD=ABS(XDAT(I,1)-XDAT(I,2))
 1498   CONTINUE
        IF(XD.GT.PI) GO TO 1499
        GO TO 1500
 1499   XD=XD-PI2
        GO TO 1498
 1500   SUMP=(XD)**2+SUMP
        SUMC=SUMC+(XDAT(I,5)-XDAT(I,4))**2
 1501   CONTINUE

      H=NO2-30
      SUMP=SUMP/H
      SUMC=SUMC/H
        XNSAMP=NSAMP
       XAA=XNSAMP+1.0
      SUMP1=(SUMP+XNSAMP*SUMP1)/XAA
      SUMP2=(SUMC+XNSAMP*SUMP2)/XAA
      DSUMP1=ALOG10(SUMP1)*10.
      DSUMP2=ALOG10(SUMP2)*10.
      IF(MOD(NSAMP,KPRNT).EQ.0)WRITE(IDEV,1508)
```

```
0205      1508 FORMAT('0',5X,'NONLINEAR CYCLIC ESTIMATOR')
0206           IF(MOD(NSAMP,KPRNT).EQ.0)WRITE(IDEV,1511)SUMP1,DSUMP1,SUMP2,
              *DSUMP2
0207      1511 FORMAT(2('0','AVERAGE STATISTICAL VARIANCE =',1PE13.6, /1X,
              * 'AVERAGE COMPUTED VARIANCE =',1PE13.6//))
0208           NSAMP=NSAMP+1
0209           ISAMP=ISAMP+1
          C
          C BUILD RESTART FILES
          C
          C FIRST COPY THE CURRENT PRIMARY FILE TO THE BACKUP
          C
0210                 OPEN(UNIT=1,NAME='RESTART.DAT',TYPE='OLD',ERR=7000)
0211                 READ(1,9998,ERR=7010)I1,I2,R1,R2,I3,R3,R4,R5,
                X    I4,I5,I6,I7,I8,I9,
                X    I10,I11,I12,I13,I14,I15,I16,
                X    R7,R8,R9,R10,R11,R12,
                X    MYRSTR
0212                 CLOSE(UNIT=1)
0213                 IF(MYRSTR.NE.12345)GO TO 7020
          C WRITE BACKUP FILE
0214                 OPEN(UNIT=1,NAME='BACKUP.DAT',TYPE='UNKNOWN',ERR=7030)
0215                 WRITE(1,9990)I1,I2,R1,R2,I3,R3,R4,R5,
                X    I4,I5,I6,I7,I8,I9,
                X    I10,I11,I12,I13,I14,I15,I16,
                X    R7,R8,R9,R10,R11,R12
0216                 CLOSE(UNIT=1)
          C WRITE CURRENT DATA ON NEW PRIMARY FILE
0217                 OPEN(UNIT=1,NAME='RESTART.DAT',TYPE='UNKNOWN',ERR=7040)
0218                 WRITE(1,9990)ISAMP,NSAMP,SUMP1,SUMP2,JGAUSS,DZZZ1,
                X    XZZZ,INTRNL,IBNF,TBNF
          C
0219                 CLOSE(UNIT=1)
          C IF WE GET HERE WE SUCCESSFULLY SET UP RESTART-SKIP AROUND ERRORS
0220                 GO TO 7050
          C
          C RESTART SETUP ERRORS
          C
          C UNABLE TO OPEN CURRENT RESTART FILE TO BUILD BACKUP
          C
0221      7000       CONTINUE
0222                 CALL ERROR(2,1)
          C END OF FILE ON RESTART FILE
0223      7010       CONTINUE
0224                 CALL ERROR(2,2)
          C RESTART FILE CONSISTENCY VALUE BAD
0225      7020       CONTINUE
0226                 CALL ERROR(2,3)
          C OPEN FAILURE ON BACKUP FILE
0227      7030       CONTINUE
0228                 CALL ERROR(2,4)
          C OPEN FAILURE ON SECOND OPEN OF RESTART FILE
0229      7040       CONTINUE
0230                 CALL ERROR(2,5)
0231      7050       CONTINUE
          C
```

```
FORTRAN IV-PLUS V02-51D                   11:10:03    05-APR-79          PAGE 7
NDRV3D.FTN         /TR:BLOCKS/WR

        C SEE IF WE QUIT DUE TO EVENT FLAG
        C
                  CALL KILLME(MYFLAG)
                  IF (ISAMP.LE.NO3) GO TO 100
        C
        C **************************** END FINISH PATH ************************
        C
        C **************************** FINISH RUN ****************************
        C
         2200 WRITE(IDEV,2201)
         2201 FORMAT ('0',10X,'NORMAL COMPLETION')
              STOP
              END
```

```
PROGRAM SECTIONS

NUMBER   NAME      SIZE                ATTRIBUTES

   1    $CODE1   006546  1715          RW,I,CON,LCL
   2    $PDATA   000114    38          RW,D,CON,LCL
   3    $IDATA   001546   435          RW,D,CON,LCL
   4    $VARS    026530  5804          RW,D,CON,LCL
   5    $TEMPS   000010     4          RW,D,CON,LCL
   6    PRINTC   000010     4          RW,D,OVR,GBL
   7    .$$$$.   014510  3236          RW,D,OVR,GBL
   8    GN       000016     7          RW,D,OVR,GBL
   9    GSEED    000014     6          RW,D,OVR,GBL
  10    BFINT    000046    19          RW,D,OVR,GBL


VARIABLES

 NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS

 AADLT  I*2   7-014462    AAM1   I*2   7-014472    AAM2   I*2   7-014474
 AANEW  I*2   7-014446    AAOLD  I*2   7-014444    AA2R   I*2   7-014456
 ACLF   I*2   7-014470    ACOS   R*4   4-026322    AGA    I*2   7-014464
 AGONEW I*2   7-014440    AGOOLD I*2   7-014434    ALDLT  R*4   7-014410
 ALF    R*4   7-000026    ALP110 R*4   4-026150    AMFAK  R*4   4-026344
 AMNEW  I*2   7-014442    AMOLD  I*2   7-014436    AM1    R*4   7-014342
 ANORM  I*2   7-014502    ASC1   I*2   7-014450    ASC2   I*2   7-014452
 ASC3   I*2   7-014454    ASIN   R*4   4-026326    ASJ    I*2   7-014504
 ASS    I*2   7-014506    AXJ    I*2   7-014500    AXP1   I*2   7-014460
 AXP2   I*2   7-014466    AZ     I*2   7-014372    AZJ    I*2   7-014476
 A11    R*4   7-000006    A22    R*4   7-000012    A33    R*4   4-026224
 CEIL   I*2   7-014432    CHAT   R*4   7-000112    CONST  R*4   7-000036
 COSF   I*2   7-014424    COSFZ  I*2   7-014356    DELF   R*4   4-026154
 DELT   R*4   7-000032    DELZ   I*2   7-014362    DEVQ2  R*4   4-026254
 DEVQ3  R*4   4-026260    DEV1   R*4   4-026234    DEV2   R*4   4-026240
 DEV3   R*4   4-026244    DEV4   R*4   4-026250    DSUMP1 R*4   4-026410
 DSUMP2 R*4   4-026414    DZZZ1  R*4   8-000000    FTC    R*4   4-026204
 G      R*4   4-026340    GA     R*4   7-014414    GONEW  R*4   7-000056
 GOOLD  R*4   7-000062    H      I*2   4-026120    I      I*2   4-026352
 IDEV   I*2   4-026146    INBUFZ I*2   7-014400    INFLAG I*2   7-014346
 IPRNT  I*2   6-000000    ISAMP  I*2   4-026266    ITOPS  I*2   7-014406
 IY2    I*2   4-026264    I1     I*2   4-026420    I10    I*2   4-026466
 I11    I*2   4-026470    I12    I*2   4-026472    I13    I*2   4-026474
 I14    I*2   4-026476    I15    I*2   4-026500    I16    I*2   4-026502
 I2     I*2   4-026422    I3     I*2   4-026434    I4     I*2   4-026452
 I5     I*2   4-026454    I6     I*2   4-026456    I7     I*2   4-026460
 I8     I*2   4-026462    I9     I*2   4-026464    J      I*2   4-026354
 JGAUSS I*2   8-000004    JNSZ   I*2   7-014364    JPRNT  I*2   6-000002
 JSEED  I*2   4-026304    JZZ    I*2   7-014366    K      I*2   4-026332
 KBIAS  I*2   7-014430    KMAX   I*2   7-000004    KOUNT  I*2   6-000006
 KPRNT  I*2   6-000004    LCHAT  I*2   7-014350    LSHAT  I*2   7-014352
 L1     I*2   4-026356    L2     I*2   4-026360    L4     I*2   4-026362
 M      I*2   7-000000    MEMS   I*2   7-014370    MN     I*2   4-026350
 MNEW   R*4   7-000046    MOLD   R*4   7-000052    MYFLAG I*2   4-026144
 MYRSTR I*2   4-026302    N      I*2   7-000002    NORM   R*4   7-000126
 NO2    I*2   4-026164    NO3    I*2   4-026166    NSAMP  I*2   4-026270
 PI      R*4   7-000066  PIDLT  R*4   7-000022  PI2    R*4   4-026230
 P110    R*4   4-026170  P220   R*4   4-026214  P330   R*4   4-026220
 QQ      R*4   4-026174  Q22    R*4   4-026210  Q22C   R*4   4-026160
 Q33     R*4   7-014420  Q33C   R*4   7-000016  RX     R*4   4-026200
 R1      R*4   4-026424  R10    R*4   4-026520  R11    R*4   7-000042
 R12     R*4   4-026524  R2     R*4   4-026430  R3     R*4   4-026436
 R4      R*4   4-026442  R5     R*4   4-026446  R7     R*4   4-026504
 R8      R*4   4-026510  R9     R*4   4-026514  SHAT   R*4   7-000116
 SINF    I*2   7-014426  SINFZ  I*2   7-014360  SN1Z   I*2   7-014354
 SUMC    R*4   4-026370  SUMP   R*4   4-026364  SUMP1  R*4   4-026272
 SUMP2   R*4   4-026276  S1Z    I*2   7-014374  S2Z    I*2   7-014376
 TWOPI   R*4   7-000072  T1Z    I*2   7-014402  T2Z    I*2   7-014404
 XAA     R*4   4-026404  XD     R*4   4-026374  XHAT   R*4   7-000122
 XNSAMP  R*4   4-026400  X1     R*4   4-026306  X2     R*4   4-026312
 X3      R*4   4-026316  YY3    R*4   4-026334  Y1EST  R*4   7-000076
 Y2EST   R*4   7-000102  Y3EST  R*4   7-000106  Z1     R*4   7-014132
 Z2      R*4   7-014136


ARRAYS

 NAME    TYPE  ADDRESS          SIZE           DIMENSIONS

 COSY    R*4   7-014142  000100    32          (16)
 IBNF    I*2  10-000000  000016     7          (7)
 INTRNL  I*2   9-000000  000014     6          (6)
 J0      R*4   7-000132  014000  3072          (1536)
 J00     R*4   4-000000  014000  3072          (1536)
 MYDATE  L*1   4-026122  000011     4          (9)
 MYTIME  L*1   4-026133  000010     4          (8)
 SINY    R*4   7-014242  000100    32          (16)
 TBNF    R*4  10-000016  000030    12          (6)
 XDAT    R*4   4-014000  012120  2600          (130,10)
 XZZZ    R*4   8-000006  000010     4          (2)


LABELS

 LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

 5       1-004026        10         **           11         **
 12      1-003326        100     1-003014        201'    3-000542
 205'       **           450     1-003726        651'    3-000032
 1498    1-004476        1499    1-004524        1500    1-004552
 1501       **           1508'   3-000600        1511'   3-000642
 2200       **           2201'   3-000764        5000    1-002466
 5010    1-002504        5015    1-002522        5020    1-002540
 6000    1-002556        7000    1-006360        7010    1-006376
 7020    1-006414        7030    1-006432        7040    1-006450
 7050    1-006466        9990'   3-000204        9991'   3-000146
 9998'   3-000066        9999'   3-000000


FUNCTIONS AND SUBROUTINES REFERENCED

 APPUT   APWD    CLOS$   CLREF   DATE    ERROR   GAUSS   GLOBAL  KILLME  LEAF
 OPEN$    TIME     $ALG10  $COS     $EXP     $SIN     $SQRT


TOTAL SPACE ALLOCATED = 054010 11268
```

```
      SUBROUTINE GAUSS(JS,SD,XM,X)
      COMMON/GSEED/  NST(6)
      COMMON /GN/ TWOPI, J, XR(2)
      IF (J) 10, 10, 20
   10 J=2
      TWOPI=8.*(ATAN(1.))
      NST(1)=25
      NST(2)=8
      NST(3)=31
      NST(4)=45
      NST(5)=20
      NST(6)=17
      XR(1)=BANF(NST,1)
      GO TO 35
   20 GO TO (30,40), J
   30 J=2
      XR(1)=BANF(NST,0)
   35 XR(2)=BANF(NST,0)
      X1=SQRT(ABS(-2.*ALOG(XR(1))))
      XR(2)=TWOPI*XR(2)
      XR(1)=X1*SIN(XR(2))
      XR(2)=X1*COS(XR(2))
      X=XR(1)*SD+XM
      RETURN
   40 J=1
      X=XR(2)*SD+XM
      RETURN
      END
```

```
FORTRAN IV-PLUS V02-51D                      11:10:42    05-APR-79              PAGE 12
NDRV3D.FTN      /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME        SIZE                   ATTRIBUTES

  1     $CODE1  000454   150                RW,I,CON,LCL
  2     $PDATA  000016     7                RW,D,CON,LCL
  3     $IDATA  000014     6                RW,D,CON,LCL
  4     $VARS   000004     2                RW,D,CON,LCL
  6     GSEED   000014     6                RW,D,OVR,GBL
  7     GN      000016     7                RW,D,OVR,GBL


ENTRY POINTS

 NAME    TYPE  ADDRESS      NAME   TYPE  ADDRESS      NAME   TYPE  ADDRESS

 GAUSS         1-000000


VARIABLES

 NAME    TYPE  ADDRESS      NAME   TYPE  ADDRESS      NAME   TYPE  ADDRESS

 J       I*2   7-000004     JS     I*2   F-000002*    SD     R*4   F-000004*
 TWOPI   R*4   7-000000     X      R*4   F-000010*    XM     R*4   F-000006*
 X1      R*4   4-000000


ARRAYS

 NAME    TYPE  ADDRESS           SIZE            DIMENSIONS

 NST     I*2   6-000000  000014      6           (6)
 XR      R*4   7-000006  000010      4           (2)


LABELS

 LABEL   ADDRESS             LABEL   ADDRESS             LABEL   ADDRESS

 10         **               20      1-000152            30      1-000174
 35      1-000234            40      1-000414


FUNCTIONS AND SUBROUTINES REFERENCED

 BANF    $ALOG    $ATAN    $COS     $SIN     $SQRT


TOTAL SPACE ALLOCATED = 000544    178
```

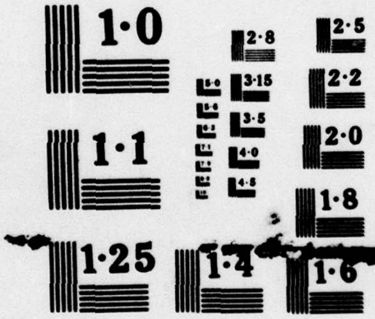
```
      FUNCTION BANF(NS,MODE)
      DIMENSION  NS(6), NC(6)
C COMMON FOR HOLDING RESTART VALUES
        COMMON/BFINT/IBNF(7),TBNF(6)
        EQUIVALENCE(N1,IBNF(1)),(N2,IBNF(2)),(N3,IBNF(3))
        EQUIVALENCE(N4,IBNF(4)),(N5,IBNF(5)),(N6,IBNF(6))
        EQUIVALENCE(MP,IBNF(7))
        EQUIVALENCE(T1,TBNF(1)),(T2,TBNF(2)),(T3,TBNF(3))
        EQUIVALENCE(T4,TBNF(4)),(T5,TBNF(5)),(T6,TBNF(6))
      DATA M1,M2,M3,M4,M5,M6/59,47,62,38,45,23/
C        MODE=0 TO CONTINUE, OTHERWISE RESTART WITH
C        INTEGER NUMBER NS(1)*2**18+NS(2)
      IF (MODE) 10, 100, 10
   10 N1=NS(1)
      N2=NS(2)
      N3=NS(3)
      N4=NS(4)
      N5=NS(5)
      N6=NS(6)
      T1=2.**(-6)
      T2=2.**(-12)
      T3=2.**(-18)
      T4=2.**(-24)
      T5=2.**(-30)
      T6=2.**(-36)
      MP=2**6
  100 DO 200 I=1,6
      GO TO (110,120,130,140,150,160),I
  110 K=N6*M6
      GO TO 190
  120 K=N6*M5+N5*M6+KD
      GO TO 190
  130 K=N6*M4+N5*M5+N4*M6+KD
      GO TO 190
  140 K=N6*M3+N5*M4+N4*M5+N3*M6+KD
      GO TO 190
  150 K=N6*M2+N5*M3+N4*M4+N3*M5+N2*M6+KD
      GO TO 190
  160 K=N6*M1+N5*M2+N4*M3+N3*M4+N2*M5+N1*M6+KD
  190 KD=K/MP
  200 NC(I)=K-KD*MP
      N1=NC(6)
      N2=NC(5)
      N3=NC(4)
      N4=NC(3)
      N5=NC(2)
      N6=NC(1)
      XN1=N1
      XN2=N2
      XN3=N3
      XN4=N4
      XN5=N5
      XN6=N6
      BANF=XN1*T1+XN2*T2+XN3*T3+XN4*T4+XN5*T5+XN6*T6
      RETURN
```

```
0053          END
```

```
FORTRAN IV-PLUS V02-51D                       11:10:47    05-APR-79              PAGE 15
NDRV3D.FTN      /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME       SIZE                       ATTRIBUTES

  1      $CODE1  001344    370                 RW,I,CON,LCL
  2      $PDATA  000016      7                 RW,D,CON,LCL
  3      $IDATA  000016      7                 RW,D,CON,LCL
  4      $VARS   000066     27                 RW,D,CON,LCL
  6      BFINT   000046     19                 RW,D,OVR,GBL


ENTRY POINTS

 NAME    TYPE  ADDRESS      NAME    TYPE  ADDRESS       NAME    TYPE  ADDRESS

 BANF    R*4   1-000000


VARIABLES

 NAME    TYPE  ADDRESS      NAME    TYPE  ADDRESS       NAME    TYPE  ADDRESS

 I       I*2   4-000030     K       I*2   4-000032      KD      I*2   4-000034
 MODE    I*2   F-000004*    MP      I*2   6-000014      M1      I*2   4-000014
 M2      I*2   4-000016     M3      I*2   4-000020      M4      I*2   4-000022
 M5      I*2   4-000024     M6      I*2   4-000026      N1      I*2   6-000000
 N2      I*2   6-000002     N3      I*2   6-000004      N4      I*2   6-000006
 N5      I*2   6-000010     N6      I*2   6-000012      T1      R*4   6-000016
 T2      R*4   6-000022     T3      R*4   6-000026      T4      R*4   6-000032
 T5      R*4   6-000036     T6      R*4   6-000042      XN1     R*4   4-000036
 XN2     R*4   4-000042     XN3     R*4   4-000046      XN4     R*4   4-000052
 XN5     R*4   4-000056     XN6     R*4   4-000062


ARRAYS

 NAME    TYPE  ADDRESS          SIZE           DIMENSIONS

 IBNF    I*2   6-000000  000016      7         (7)
 NC      I*2   4-000000  000014      6         (6)
 NS      I*2   F-000002* 000014      6         (6)
 TBNF    R*4   6-000016  000030     12         (6)


LABELS

 LABEL   ADDRESS            LABEL   ADDRESS             LABEL   ADDRESS

 10          **             100     1-000334            110     1-000372
 120     1-000420           130     1-000462            140     1-000536
 150     1-000624           160     1-000724            190     1-001034
 200         **


TOTAL SPACE ALLOCATED = 001534    430
```

```
      SUBROUTINE GLOBAL
C      VERSION 5/5/1978
        REAL A(20),ABOX(20),J0(1536),SIGMA(16),NORM,MOLD,MNEW,
     1 S1(16),S2(16),PSI(96),DELJ(96)
        INTEGER SN1Z,COSFZ,SINFZ,DELZ,AZ,S1Z,S2Z,T1Z,T2Z,JNS(96)
        INTEGER COSF,SINF,CEIL,AGOOLD,AMOLD,AGONEW,AMNEW
        INTEGER AAOLD,AANEW,ASC1,ASC2,ASC3,AA2R,AXP1,AADLT
        INTEGER ASS
        INTEGER AXP2,AGA,ACLF,AAM1,AAM2,AZJ,AXJ,ANORM,ASJ

        COMMON M,N,KMAX,A11,A22,Q33C,PIDLT,ALF,DELT,CONST,R11,
     1 MNEW,MOLD,GONEW,GOOLD,PI,TWOPI,Y1EST,Y2EST,Y3EST,
     2 CHAT,SHAT,XHAT,NORM,J0,Z1,Z2,
     3 COSY(16),SINY(16),AM1
        COMMON INFLAG,LCHAT,LSHAT,SN1Z,COSFZ,SINFZ,DELZ,JNSZ,JZZ,
     1  MEMS,AZ,S1Z,S2Z,INBUFZ,T1Z,T2Z,ITOPS,ALDLT,GA,Q33,COSF,
     2  SINF,KBIAS,CEIL,AGOOLD,AMOLD,AGONEW,AMNEW,AAOLD,AANEW,ASC1,
     3  ASC2,ASC3,AA2R,AXP1,AADLT,AGA,AXP2,ACLF,AAM1,AAM2,AZJ,AXJ,
     4  ANORM,ASJ,ASS
C     GLOBAL INITIALIZATIONS FOR NONLINEAR FILTER
 200  CALL APINIT(1,1,III)
C     CLEAR MD(0)-MD(8191)
        DO 202 I=1,1024
 202  J0(I)=0.0
        ISTART=0
      DO 204 I=1,64
      CALL APPUT(J0,ISTART,1024,2)
        ISTART=ISTART+1024
 204   CONTINUE
        A5=-1./R11/2.
        CALL APPUT(A5,AXP1,1,2)
        CALL APWD
        X5=DELT*ALF
        X5=-X5
        CALL APPUT(X5,AADLT,1,2)
        CALL APWD
        A5=-GA
        CALL APPUT(A5,AGA,1,2)
        CALL APWD
        A5=-1./2./Q33C/DELT
        CALL APPUT(A5,AXP2,1,2)
        CALL APWD
        A5=1.
        CALL APPUT(A5,ANORM,1,2)
        CALL APWD
C
        NORM=1.0
C
C     PHASE VARIABLES
      DO 210 I=1,M
      SIGMA(I)=PI*((2.*I-1.)/FLOAT(M)-1.)
      COSY(I)=COS(SIGMA(I))
      SINY(I)=SIN(SIGMA(I))
      S1(I)=COSY(I)/R11
 210  S2(I)=SINY(I)/R11
      CALL APPUT(COSY,COSFZ,M,2)
```

AD-A073 051   UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES DEPT O--ETC F/G 9/2
SOFTWARE FOR NONLINEAR FILTERING.(U)
JUN 79 R S BUCY, F GHOVANLOU   F44620-76-C-0085
UNCLASSIFIED   USCAE-53-4514-1787   AFOSR-TR-79-0923   NL


1·0
2·8
2·5
5·0
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

```
      CALL APPUT(SINY,SINFZ,M,2)
      CALL APPUT(S1,S1Z,M,2)
      CALL APPUT(S2,S2Z,M,2)
       CALL APWD
C
C     PHASE RATE VARIABLES
      DO 220 I=1,N
  220  PSI(I)=PIDLT*((2.*I-1.)/FLOAT(N)-1.)
C
C   INTERPOLATION ADDRESS AND FACTORS
       AM=M
       AN=N
       DO 230 J=1,N
       AJ=J
       PRQ=(AM/AN+AM)/2.-AM/AN*AJ+AM
       IRQ=PRQ
       DELJ(J)=PRQ-IRQ
  230   JNS(J)=MOD(IRQ,M)+1
      CALL APPUT(DELJ,DELZ,N,2)
       CALL APPUT(JNS,JNSZ,N,1)
      CALL APWD
C
C     EVALUATE CONVOLUTION TERMS A(I)
      DO 280 I=1,NTERM
      TEMP=I/FLOAT(N)
      TEMP=CONST*TEMP*TEMP
C     A(I)=0.
C280  IF (TEMP.GT.-20) A(I)=EXP(TEMP)
  280 A(I)=EXP(TEMP)
      DO 282 I=1,5
  282 ABOX(I)=A(6-I)
      ABOX(6)=1.
      DO 284 I=1,5
  284 ABOX(I+6)=A(I)
       ABOX(1)=0.0
       ABOX(11)=0.0
      CALL APPUT(ABOX,AZ,11,2)
      CALL APWD
C     DUMPED ABOX HERE
C     CONSTRUCT THE A PRIORI DENSITY
C     CNORM=1.0/(TWOPI*SQRT(A11*A22))
       CNORM=1.
      CL=-0.5/A22
      SI=-0.5/A11
      DO 290 I=1,M
      CR=SIGMA(I)-Y1EST
      CR=CR*CR*SI
      J1=0
      DO 290 J=1,N
      J2=J1+I
      TEMP=PSI(J)-Y2EST
       JO(J2)=0.
       TEMP1=TEMP*TEMP*CL+CR
       IF (TEMP1.LE.-27) GOTO 290
     JO(J2)=EXP(TEMP1)*CNORM
  290 J1=J1+M
```

```
FORTRAN IV-PLUS V02-51D                        12:25:14    05-APR-79           PAGE 3
GLOBAL.FTN        /TR:BLOCKS/WR

0087             RETURN
0088               END
```

```
FORTRAN IV-PLUS V02-51D                          12:25:14    05-APR-79      PAGE 4
GLOBAL.FTN      /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME        SIZE                        ATTRIBUTES

  1     $CODE1   002120    552                   RW,I,CON,LCL
  2     $PDATA   000020      8                   RW,D,CON,LCL
  3     $IDATA   000214     70                   RW,D,CON,LCL
  4     $VARS    002540    688                   RW,D,CON,LCL
  5     $TEMPS   000014      6                   RW,D,CON,LCL
  6     .$$$$.   014510   3236                   RW,D,OVR,GBL


ENTRY POINTS

 NAME    TYPE  ADDRESS      NAME    TYPE  ADDRESS      NAME    TYPE  ADDRESS

 GLOBAL        1-000000


VARIABLES

 NAME    TYPE  ADDRESS      NAME    TYPE  ADDRESS      NAME    TYPE  ADDRESS

 AADLT   I*2   6-014462     AAM1    I*2   6-014472     AAM2    I*2   6-014474
 AANEW   I*2   6-014446     AAOLD   I*2   6-014444     AA2R    I*2   6-014456
 ACLF    I*2   6-014470     AGA     I*2   6-014464     AGONEW  I*2   6-014440
 AGOOLD  I*2   6-014434     AJ      R*4   4-002470     ALDLT   R*4   6-014410
 ALF     R*4   6-000026     AM      R*4   4-002456     AMNEW   I*2   6-014442
 AMOLD   I*2   6-014436     AM1     R*4   6-014342     AN      R*4   4-002462
 ANORM   I*2   6-014502     ASC1    I*2   6-014450     ASC2    I*2   6-014452
 ASC3    I*2   6-014454     ASJ     I*2   6-014504     ASS     I*2   6-014506
 AXJ     I*2   6-014500     AXP1    I*2   6-014460     AXP2    I*2   6-014466
 AZ      I*2   6-014372     AZJ     I*2   6-014476     A11     R*4   6-000006
 A22     R*4   6-000012     A5      R*4   4-002446     CEIL    I*2   6-014432
 CHAT    R*4   6-000112     CL      R*4   4-002514     CNORM   R*4   4-002510
 CONST   R*4   6-000036     COSF    I*2   6-014424     COSFZ   I*2   6-014356
 CR      R*4   4-002524     DELT    R*4   6-000032     DELZ    I*2   6-014362
 GA      R*4   6-014414     GONEW   R*4   6-000056     GOOLD   R*4   6-000062
 I       I*2   4-002442     III     I*2   4-002440     INBUFZ  I*2   6-014400
 INFLAG  I*2   6-014346     IRQ     I*2   4-002500     ISTART  I*2   4-002444
 ITOPS   I*2   6-014406     J       I*2   4-002466     JNSZ    I*2   6-014364
 JZZ     I*2   6-014366     J1      I*2   4-002530     J2      I*2   4-002532
 KBIAS   I*2   6-014430     KMAX    I*2   6-000004     LCHAT   I*2   6-014350
 LSHAT   I*2   6-014352     M       I*2   6-000000     MEMS    I*2   6-014370
 MNEW    R*4   6-000046     MOLD    R*4   6-000052     N       I*2   6-000002
 NORM    R*4   6-000126     NTERM   I*2   4-002502     PI      R*4   6-000066
 PIDLT   R*4   6-000022     PRQ     R*4   4-002474     Q33     R*4   6-014420
 Q33C    R*4   6-000016     R11     R*4   6-000042     SHAT    R*4   6-000116
 SI      R*4   4-002520     SINF    I*2   6-014426     SINFZ   I*2   6-014360
 SN1Z    I*2   6-014354     S1Z     I*2   6-014374     S2Z     I*2   6-014376
 TEMP    R*4   4-002504     TEMP1   R*4   4-002534     TWOPI   R*4   6-000072
 T1Z     I*2   6-014402     T2Z     I*2   6-014404     XHAT    R*4   6-000122
 X5      R*4   4-002452     Y1EST   R*4   6-000076     Y2EST   R*4   6-000102
 Y3EST   R*4   6-000106     Z1      R*4   6-014132     Z2      R*4   6-014136
```

```
FORTRAN IV-PLUS V02-51D                  12:25:14    05-APR-79          PAGE 5
GLOBAL.FTN        /TR:BLOCKS/WR

ARRAYS

 NAME    TYPE   ADDRESS        SIZE          DIMENSIONS

 A       R*4    4-000000  000120     40      (20)
 ABOX    R*4    4-000120  000120     40      (20)
 COSY    R*4    6-014142  000100     32      (16)
 DELJ    R*4    4-001340  000600    192      (96)
 JNS     I*2    4-002140  000300     96      (96)
 J0      R*4    6-000132  014000   3072      (1536)
 PSI     R*4    4-000540  000600    192      (96)
 SIGMA   R*4    4-000240  000100     32      (16)
 SINY    R*4    6-014242  000100     32      (16)
 S1      R*4    4-000340  000100     32      (16)
 S2      R*4    4-000440  000100     32      (16)


LABELS

 LABEL     ADDRESS            LABEL    ADDRESS            LABEL    ADDRESS

 200         **               202        **               204        **
 210         **               220        **               230        **
 280         **               282        **               284        **
 290       1-002052


FUNCTIONS AND SUBROUTINES REFERENCED

 APINIT   APPUT    APWD     $COS     $EXP     $SIN


TOTAL SPACE ALLOCATED = 021640  4560

,GLOBAL=GLOBAL

>
```

```
FORTRAN IV-PLUS V02-51D             12:10:47     05-APR-79           PAGE 1
LEAF.FTN         /TR:BLOCKS/WR

        SUBROUTINE LEAF
        REAL MNEW,MOLD,NORM,AOLD(16),AKRNL(16,16),B(2),XJ(16),ZJ(16),
     1  JNEWK(16),JO(1536),ANEW(16),AJOLDK(16),AKRN(256)
        INTEGER SN1Z,SINFZ,COSFZ,DELZ,AZ,S1Z,S2Z,T1Z,T2Z
        INTEGER MEM(6),KADR(16)
        INTEGER COSF,SINF,CEIL,AGOOLD,AMOLD,AGONEW,AMNEW
        INTEGER AAOLD,AANEW,ASC1,ASC2,ASC3,AA2R,AXP1,AADLT
        INTEGER ASS
        INTEGER AXP2,AGA,ACLF,AAM1,AAM2,AZJ,AXJ,ANORM,ASJ
        REAL A(2),FMEM(6)

C THIS COMMON CONTAINS THE PRINT FLAGS-SPECIFICALLY JPRNT AND KOUNT
C DATA 'TYPE'D AT BOTTOM OF THIS ROUTINE IF  MOD(KOUNT,JPRNT) IS ZERO
        COMMON/PRINTC/IPRNT,JPRNT,KPRNT,KOUNT
        COMMON M,N,KMAX,A11,A22,Q33C,PIDLT,ALF,DELT,CONST,R11,
    1 MNEW,MOLD,GONEW,GOOLD,PI,TWOPI,Y1EST,Y2EST,Y3EST,
    2 CHAT,SHAT,XHAT,NORM,JO,Z1,Z2,
    3 COSY(16),SINY(16),AM1
        COMMON INFLAG,LCHAT,LSHAT,SN1Z,COSFZ,SINFZ,DELZ,JNSZ,JZZ,
    1  MEMS,AZ,S1Z,S2Z,INBUFZ,T1Z,T2Z,ITOPS,ALDLT,GA,Q33,COSF,
    2  SINF,KBIAS,CEIL,AGOOLD,AMOLD,AGONEW,AMNEW,AAOLD,AANEW,ASC1,
    3  ASC2,ASC3,AA2R,AXP1,AADLT,AGA,AXP2,ACLF,AAM1,AAM2,AZJ,AXJ,
    4  ANORM,ASJ,ASS
C ****************** START LEAF MODULE  ********************************
        X1=SECNDS(0.0)
        IDEV=5
        T1=SECNDS(X1)
        CALL VRAMP(AGOOLD,AMOLD,ASC1,1,KMAX)
        CALL APWR
        CALL VSADD(ASC1,1,AMOLD,AAOLD,1,KMAX)
        CALL APWR
        CALL VRAMP(AGONEW,AMNEW,ASC2,1,KMAX)
        CALL APWR
        CALL VSADD(ASC2,1,AMNEW,AANEW,1,16)
        CALL APWR
        IIII=ASS+2*M
        IV=IIII+1
        XX1=-1.
        CALL APPUT(XX1,IIII,1,2)
        CALL APWD
        CALL VSADD(AAOLD,1,IIII,ASC2,1,16)
        CALL APWR
        CALL VEXP(ASC2,1,IV,1,16)
        CALL APWR
        CALL VSQ(IV,1,ASC1,1,KMAX)
        CALL APWR
        CALL VSMUL(ASC1,1,AXP1,ASC2,1,KMAX)
        CALL APWR
        CALL VEXP(ASC2,1,AA2R,1,KMAX)
        CALL APWR
       T2=SECNDS(X1)
      JZ=ITOPS
        MEM(1)=JZ+M*N-1
        MEM(2)=MEM(1)
        MEM(3)=JZ+M*N
        MEM(4)=JZ
        MEM(5)=JZ+M*N-4*M-1
        MEM(6)=JZ-1
      B(1)=Z1
      B(2)=Z2
        CALL APPUT(B,18,2,2)
        AAA=FLOAT(KBIAS)
        II=ASS+3
        CALL APPUT(AAA,II,1,2)
        FMEM(1)=FLOAT(MEM(1))
        FMEM(2)=FLOAT(MEM(2))
        FMEM(3)=FLOAT(MEM(3))
        FMEM(4)=FLOAT(MEM(4))
        FMEM(5)=FLOAT(MEM(5))
        FMEM(6)=FLOAT(MEM(6))
        III=ASS+4
        CALL APPUT(FMEM,III,6,2)
        CALL APWD
        CALL LC(III,II,KBIAS,IV,INBUFZ,AA2R,ASC1)
        CALL APWR
        T3=SECNDS(X1)
        AB=NORM
        TNORM=1./AB
        CALL APPUT(TNORM,ANORM,1,2)
        CALL APWD
        CALL ME(AANEW,AAOLD,ASC1,AADLT,ASC2,ASC3,AGA,AXP2,ACLF,ANORM)
        CALL APWR
        T4=SECNDS(X1)
        CALL TTMOV(ACLF,4608,256)
        CALL APWR
        T5=SECNDS(X1)
C ********************* END LEAF MODULE ***********************************
        DO 600 K=1,16
  600   XJ(K)=0.0
        DO 609 K=1,16
  609   CALL APPUT(XJ,ITOPS+M*N+(K-1)*M*(N+1),16,2)
        CALL APWD
C ********************* START ACON MODULE *********************************
C         DOES AMPLITUDE CUNVOLUTION FOR EACH I,J
        T6=SECNDS(X1)
        CALL W
        CALL APWR
        T7=SECNDS(X1)
        CALL XSUM(ITOPS,M,AXJ,KBIAS)
        CALL APWR
        CALL ZSUM(ITOPS,M,AZJ,KBIAS)
        CALL APWR
        T8=SECNDS(X1)
        CALL DOTPR(AXJ,1,COSFZ,1,ASS,M)
        CALL DOTPR(AXJ,1,SINFZ,1,ASS+1,M)
        CALL DOTPR(AZJ,1,AANEW,1,AAM1,KMAX)
        CALL VSQ(AANEW,1,ASC1,1,KMAX)
        CALL DOTPR(ASC1,1,AZJ,1,AAM2,KMAX)
        CALL SVE(AZJ,1,ANORM,KMAX)
        CALL APWR
        CALL APGET(A,ASS,2,2)
        CALL APGET(Y,AAM1,1,2)
        CALL APGET(YY,AAM2,1,2)
        CALL APGET(ABL,ANORM,1,2)
        CALL APWD
        T0=SECNDS(X1)-T8
        XHAT=ATAN2(A(2),A(1))
        NORM=ABL
        AM1=Y/ABL
        AM2=YY/ABL
        CALL VMOV(AGONEW,1,AGOOLD,1,2)
        CALL APWR
     S1=AMAX1(AM2-AM1**2,1.E-18)
     S1=SQRT(S1)
      GONEW=(-.5-FLOAT(KMAX)/2.)*(1./2.)*S1+AM1
        MNEW=.5*S1
        CALL APPUT(GONEW,AGONEW,1,2)
        CALL APPUT(MNEW,AMNEW,1,2)
        CALL APWD
C ****************** END ACON MODULE ********************************
        T9=SECNDS(X1)
        IF(MOD(KOUNT,JPRNT).EQ.0)TYPE *,T1,T2-T1,T3-T2,T4-T3,T5-T4,T6-T5
     X      ,T7-T6,T8-T7,T9-T8,T9
        IF(MOD(KOUNT,JPRNT).EQ.0)TYPE *,T0
      RETURN
      END
```

```
FORTRAN IV-PLUS V02-51D            12:10:47    05-APR-79         PAGE 4
LEAF.FTN         /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME         SIZE                    ATTRIBUTES

  1     $CODE1  002462   665                  RW,I,CON,LCL
  2     $PDATA  000044    18                  RW,D,CON,LCL
  3     $IDATA  000636   207                  RW,D,CON,LCL
  4     $VARS   005062  1305                  RW,D,CON,LCL
  6     PRINTC  000010     4                  RW,D,OVR,GBL
  7     .$$$$.  014510  3236                  RW,D,OVR,GBL


ENTRY POINTS

 NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS

 LEAF          1-000000


VARIABLES

 NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS     NAME    TYPE  ADDRESS

 AAA     R*4   4-004754    AADLT   I*2   7-014462    AAM1    I*2   7-014472
 AAM2    I*2   7-014474    AANEW   I*2   7-014446    AAOLD   I*2   7-014444
 AA2R    I*2   7-014456    AB      R*4   4-004770    ABL     R*4   4-005036
 ACLF    I*2   7-014470    AGA     I*2   7-014464    AGONEW  I*2   7-014440
 AGOOLD  I*2   7-014434    ALDLT   R*4   7-014410    ALF     R*4   7-000026
 AMNEW   I*2   7-014442    AMOLD   I*2   7-014436    AM1     R*4   7-014342
 AM2     R*4   4-005046    ANORM   I*2   7-014502    ASC1    I*2   7-014450
 ASC2    I*2   7-014452    ASC3    I*2   7-014454    ASJ     I*2   7-014504
 ASS     I*2   7-014506    AXJ     I*2   7-014500    AXP1    I*2   7-014460
 AXP2    I*2   7-014466    AZ      I*2   7-014372    AZJ     I*2   7-014476
 A11     R*4   7-000006    A22     R*4   7-000012    CEIL    I*2   7-014432
 CHAT    R*4   7-000112    CONST   R*4   7-000036    COSF    I*2   7-014424
 COSFZ   I*2   7-014356    DELT    R*4   7-000032    DELZ    I*2   7-014362
 GA      R*4   7-014414    GONEW   R*4   7-000056    GOOLD   R*4   7-000062
 IDEV    I*2   4-004730    II      I*2   4-004760    III     I*2   4-004762
 IIII    I*2   4-004736    INBUFZ  I*2   7-014400    INFLAG  I*2   7-014346
 IPRNT   I*2   6-000000    ITOPS   I*2   7-014406    IV      I*2   4-004740
 JNSZ    I*2   7-014364    JPRNT   I*2   6-000002    JZ      I*2   4-004752
 JZZ     I*2   7-014366    K       I*2   4-005010    KBIAS   I*2   7-014430
 KMAX    I*2   7-000004    KOUNT   I*2   6-000006    KPRNT   I*2   6-000004
 LCHAT   I*2   7-014350    LSHAT   I*2   7-014352    M       I*2   7-000000
 MEMS    I*2   7-014370    MNEW    R*4   7-000046    MOLD    R*4   7-000052
 N       I*2   7-000002    NORM    R*4   7-000126    PI      R*4   7-000066
 PIDLT   R*4   7-000022    Q33     R*4   7-014420    Q33C    R*4   7-000016
 R11     R*4   7-000042    SHAT    R*4   7-000116    SINF    I*2   7-014426
 SINFZ   I*2   7-014360    SN1Z    I*2   7-014354    S1      R*4   4-005052
 S1Z     I*2   7-014374    S2Z     I*2   7-014376    TNORM   R*4   4-004774
 TWOPI   R*4   7-000072    T0      R*4   4-005042    T1      R*4   4-004732
 T1Z     I*2   7-014402    T2      R*4   4-004746    T2Z     I*2   7-014404
 T3      R*4   4-004764    T4      R*4   4-005000    T5      R*4   4-005004
 T6      R*4   4-005012    T7      R*4   4-005016    T8      R*4   4-005022
 T9      R*4   4-005056    XHAT    R*4   7-000122    XX1     R*4   4-004742
 X1      R*4   4-004724    Y       R*4   4-005026    YY      R*4   4-005032
 Y1EST   R*4   7-000076    Y2EST   R*4   7-000102    Y3EST   R*4   7-000106
 Z1      R*4   7-014132    Z2      R*4   7-014136


ARRAYS

 NAME    TYPE  ADDRESS        SIZE        DIMENSIONS

 A       R*4   4-004664  000010     4     (2)
 AJOLDK  R*4   4-002510  000100    32     (16)
 AKRN    R*4   4-002610  002000   512     (256)
 AKRNL   R*4   4-000100  002000   512     (16,16)
 ANEW    R*4   4-002410  000100    32     (16)
 AOLD    R*4   4-000000  000100    32     (16)
 B       R*4   4-002100  000010     4     (2)
 COSY    R*4   7-014142  000100    32     (16)
 FMEM    R*4   4-004674  000030    12     (6)
 JNEWK   R*4   4-002310  000100    32     (16)
 JO      R*4   7-000132  014000  3072     (1536)
 KADR    I*2   4-004624  000040    16     (16)
 MEM     I*2   4-004610  000014     6     (6)
 SINY    R*4   7-014242  000100    32     (16)
 XJ      R*4   4-002110  000100    32     (16)
 ZJ      R*4   4-002210  000100    32     (16)


LABELS

 LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

 600        **           609        **


FUNCTIONS AND SUBROUTINES REFERENCED

 APGET   APPUT   APWD    APWR    DOTPR   LC      ME      SECNDS  SVE     TTMOV
 VEXP    VMOV    VRAMP   VSADD   VSMUL   VSQ     W       XSUM    ZSUM    $AMAX1
 $ATAN2  $SQRT


TOTAL SPACE ALLOCATED = 025166  5435

,LEAF=LEAF

>
```

```
    DEFINE LC(III,II,KBIAS,AANEW,INBUFZ,AA2R,ASC1)
    LOCAL K,A,B
    K=0
    CALL VFIX(III,1,1,1,6)
    LOOP:A=AANEW+K
    B=AA2R+K
    CALL VSMUL(18,1,A,INBUFZ,1,2)
    CALL VMOV(B,1,7,1,1)
    CALL VSADD(III,1,II,ASC1,1,6)
    CALL VMOV(ASC1,1,III,1,6)
    CALL RLNLF(0)
    CALL VFIX(ASC1,1,1,1,6)
    K=K+1
    IF K<16 GOTO LOOP
    END

>
```

```
    "RLNLFE.FSO
       "DIC.FSO    FAST 2-LOOP BOX
        "DOES FILTER(FRANK), THEN
            "CONVOLVE(JACK).
        "INITSW=0 FIRST CALL-ONLY DOES CONVOLVE.
        "INITSW=0 FIRST FILTER CALL
              "=1 REST OF FILTER CALLS
              "=2 CONVOLVE CALL
        "INITSW=1 FOR EACH ESTIMATE.
         "REMOVE HALT BETWEEN FILTER,CONVOLVE
        "VERSION
        "3-8-78 HALT IM MIDDLE
            "RLNLEH16 TO RLNLFE3D
        "3-8-78
         "CALLS EXPDO
         "NEXTMD -ADDRESSING ENTRY FOR EXPDO
         "LD NORM FROM MD IN FRANK
          "HALT AFTER FRANK

                $TITLE RLNLF
                $ENTRY RLNLF,1      "CALL NLF(INITSW=CHECK)
                $ENTRY NEXTMD       "SP(0):=NEXT FREE MD ADDR.
                $EXT  EXPDO         "DO EXP IN BOX

     "SIZING
        MV $EQU 16.
        NV $EQU 96.
        MNV $EQU MV*NV
        SN1ZV $EQU 20.
        COSFZ $EQU SN1ZV+MV
        SINFZ $EQU COSFZ+MV
        DELZ $EQU SINFZ+MV
        JNSZZ $EQU DELZ+NV
        JZZ $EQU JNSZZ+NV
        MEMS $EQU JZZ+MNV+MV
        AZ $EQU MEMS+11.
        NXFREE $EQU AZ+11.
"TABLE MEMORY ADDRESES
        JTMA $EQU 10000
        NORLV $EQU JTMA+NV+11.
        "MN1V $EQU MNV+JZ-1
        "MN2V $EQU MNV+JZ
        XINCV $EQU 1
        M1V $EQU MV-1
        TWV $EQU 2
        MMV $EQU TWV*MV
        "MNNV $EQU MNV+JZ-1
        CHATV $EQU 18.
        SHATV $EQU 19.
        CHECKV $EQU 0
```

```
"S PAD ADDRESSES
     "GLOBAL CONSTANTS
        M $EQU 1
        N $EQU 2
        MN $EQU 3

    "FOR FILTER
        ICNT $EQU 3
        MNN $EQU 4
        JCNT $EQU 4      "C+
        MN1 $EQU 5
        JFI $EQU 5
        M2 $EQU 6
        XINC $EQU 7
        JRA $EQU 7
        JNSRA $EQU 8.
        JWA $EQU 8.
        M1 $EQU 9.
        MM $EQU 10.
        MN2 $EQU 11.
        TW $EQU 12.
        JBI $EQU 12.
        JNSZ $EQU 13.
        COSFA $EQU 13.
        NORL $EQU 14.
        CHAT $EQU 14.
        SINFA $EQU 14.
        INITSW $EQU 0     "C+.
        SN1Z $EQU 0
RLNLF:NOP
     JSR EXPDO
     JMP ENTER1

NEXTMD: LDSPI 0;DB=NXFREE      "RETURN SP(0):=NEXT FREE MD ADDR.
        RETURN
        NOP
        NOP

ENTER1:LDMA;                 "MAIN ENTRY FROM HOST
          DB=2

        NOP

        LDMA;
          DB=3

        LDSPI MNN;
          DB=MD

        LDSPI MN1;
          DB=MD

        LDSPI MN2;
          DB=MD

        NOP

        NOP
        LDSPI M;
          DB=MV

        LDDA;
          DB=5

        LDSPI N;
          DB=NV

        LDSPI MN;
          DB=MNV

        LDSPI JNSRA;
          DB=JZZ

        LDSPI XINC;
          DB=XINCV

        LDSPI M1;
          DB=M1V

        LDSPI MM;
          DB=MMV

        LDSPI TW;
          DB=TWV

        LDSPI JNSZ;
          DB=JNSZZ

        LDSPI NORL;
          DB=NORLV

        MOV INITSW,INITSW

        BEQ FIRST

        LDSPI 15;DB=2    "IF INITSW=2 GOTO CONVOL
        SUB 15,INITSW
        BNE  NXLABEL
        JMP CONVOL
  NXLABEL: JMP SECOND    "ELSE GOTO NON-FIRST FILTER

FIRST:LDDPA;
          DB=10.

      LDTMA;DB=!ONE        "TM(SP(NORL)):= 1.0
      NOP

      DPX<TM

        MOV NORL,NORL;
          SETTMA;
          OUT;
          DB=DPX

        LDTMA;
          DB=JTMA+10.
        LDMA;
          DB=DELZ

        "STO N DELJ IN TM
        MOV N,JCNT

LP1:INCMA
        INCTMA;
          OUT;
          DB=MD

        DEC JCNT

        BGT LP1

        LDSPI SN1Z;DB=SN1ZV

        JMP ST1      "DO ONLY CONVOLVE FIRST TIME
                  "DO FRANK

SECOND:LDDPA;          "BEGIN FILTER
          DB=10.

        LDSPI 0;DB=CHATV-1       "ADDR OF NORM
        MOV 0,0;SETMA
        NOP
        NOP
        MOV NORL,NORL;
          SETTMA;
          OUT;
          DB=MD      "NORM

ST1:LDMA;                      "NORM*EXP(-A**2/2*R)--> NORM
          DB=7

        LDDPA;
          DB=0

        LDTMA;
          DB=!ONE
        "MOV NORL,NORL;
          "SETTMA

     LDMA;DB=2

       DPX<MD

       NOP

       FMUL TM,DPX

     LDSPI MNN;DB=MD

       FMUL

       FMUL
```

```
        DPX<FM

        MOV NORL,NORL;
          SETTMA;
          OUT;
          DB=DPX

        "STO M SN IN DPX & DPY

LPCYC:LDSPI SN1Z;
          DB=SN1ZV

LP2:MOV SN1Z,SN1Z;
        SETMA

        MOV NORL,NORL;
          SETTMA

        INC SN1Z

        FMUL TM,MD

        FMUL

        FMUL;
          DEC M

        DPX(0)<FM,DPY(0)<FM

        INCDPA

        BGT LP2

        LDSPI M;
          DB=MV

        DEC TW

        BGT LPCYC

        LDSPI TW;
          DB=TWV

        LDTMA;
          DB=JTMA+NV+11.
        LDSPI M;
          DB=MV

LOOP:DEC JNSRA
        MOV JNSRA,JNSRA;
          SETMA

        NOP

        NOP

        LDSPI JNSZ;
          DB=MD
          DECTMA

          AND# XINC,JNSZ

          BEQ BIELP

          JMP BIOLP

BIELP:ADD# XINC,JNSZ;
          SETDPA

          SUB 1,5

          SUB 1,11.

          ADD# MN1,JNSZ;
            SETMA                          "GET J1

          INC JNSZ

          MOVR 1,6                         "M2=16.

          AND M1,JNSZ;
            FMUL DPX(-2),MD                "J1*SN1--->J1

          ADD# MN1,JNSZ;                   "GET J2
            SETMA;
            FMUL

          FMUL

          ADD# MN2,JNSZ;                   "GET J3
            SETMA;
            DPY(-2)<FM                     "STO J1 IN DPY 0

          ADD TW,JNSZ;
            FMUL DPY(-1),MD                "J2*SN2--->J2

          FMUL;
            INCDPA

          FMUL DPX(-1),MD                  "J3*SN3--->J3

          FMUL;
            DPX(-2)<FM;                    "STO J2 IN DPX 1
            INCDPA

          AND M1,JNSZ;
            FMUL

          ADD# MN1,JNSZ;                   "GET J4
            SETMA;
            DPY(-2)<FM                     "STO J3 IN DPY 2

          NOP

          ADD# MN2,JNSZ;                   "GET J5
            SETMA

          ADD TW,JNSZ;
            FMUL DPY(-1),MD                "J4*SN4--->J4

          FMUL
```

```
        FMUL DPX(0),MD                  "J5*SN5--->J5

        FSUB DPX(-3),DPY(-4)            "J2-J1

        FMUL;
          DPX(-1)<FM;                   "STO J4 IN DPX 3
          FADD

IELP:AND M1,JNSZ;                       "CHECK IF JNSZ IS 32
          FMUL TM,FA;                   "DELJ*(J2-J1)
          FSUB DPY(-2),DPX(-3)          "J3-J2

        ADD# MN1,JNSZ;                  "GET J6
          SETMA;
          DPY(0)<FM;                    "STO J5 IN DPY 4
          FMUL;
          FADD

        FMUL TM,FA                      "DELJ*(J3-J2)

        ADD# MN2,JNSZ;                  "GET J7
          SETMA;
          FADD FM,DPY(-4);              "DELJ*(J2-J1)+J1
          FMUL

        ADD TW,JNSZ;
          FMUL DPY(1),MD;               "J6*SN6--->J6
          FADD

        FMUL;
          FADD FM,DPX(-3);              "DELJ*(J3-J2)+J2
          INCDPA;
          INC MNN;
          SETMA;
          MI<FA

        FMUL DPX(1),MD

        FSUB DPX(-2),DPY(-3);           "J4-J3
          DEC M2

        FMUL;
          DPX(0)<FM;                    "STO J6 IN DPX 5
          INCDPA;
          FADD;
          INC MNN;
          SETMA;
          MI<FA;
          BGT IELP

        SUB 10.,4

        JMP STEP

BIOLP:ADD# XINC,JNSZ;
          SETDPA
```

```
      SUB 1,5

      SUB 1,11.

      ADD# MN1,JNSZ;
        SETMA

      INC JNSZ

      MOVR 1,6

      ADD# MN1,JNSZ;
        SETMA;
        FMUL DPY(-2),MD

      AND M1,JNSZ

        ADD# MN2,JNSZ;
        SETMA;
        FMUL

      ADD TW,JNSZ;
        FMUL DPX(-1),MD

      FMUL;
        DPX(-2)<FM;
        INCDPA

      FMUL DPY(-1),MD

      FMUL;
        DPY(-2)<FM;
        INCDPA

      FMUL

      DPX(-2)<FM;
        ADD# MN1,JNSZ;
        SETMA

      AND M1,JNSZ

      ADD# MN2,JNSZ;
        SETMA

      ADD TW,JNSZ;
        FMUL DPX(-1),MD

      FMUL;
        FSUB DPY(-3),DPX(-4)

      FMUL DPY(0),MD

      FMUL;
        DPY(-1)<FM;
        FADD

IOLP:FMUL TM,FA;
        FSUB DPX(-2),DPY(-3)
        ADD# MN1,JNSZ;
          SETMA;
          DPX(0)<FM;
          FADD;
          FMUL

        AND M1,JNSZ;
          FMUL TM,FA

        ADD# MN2,JNSZ;
          SETMA;
          FADD FM,DPX(-4);
          FMUL

        ADD TW,JNSZ;
          FMUL DPX(1),MD;
          FADD

        FMUL;
          FADD FM,DPY(-3);
          INCDPA;
          INC MNN;
          SETMA;
          MI<FA

        FMUL DPY(1),MD

          FSUB DPY(-2),DPX(-3);
          DEC M2

        FMUL;
          DPY(0)<FM;
          INCDPA;
          FADD;
          INC MNN;
          SETMA;
          MI<FA;
          BCT IOLP

         SUB 10.,4

STEP:DEC N

        BEQ ASTEP
        JMP LOOP        "END MAIN FILTER LOOP

        ASTEP:NOP

  CONVOL:  NOP         "BEGIN CONVOLVE
      NOP


        LDSPI COSFA; DB=COSFZ+MV
        LDSPI SINFA;DB=SINFZ+MV
  "NORM IS READ INTO MA 10 EACH TIME
  "START CONVOLUTION LOOP
        LDSPI M;DB=MV
        LDSPI N;DB=NV
        LDDPA;
          DB=9.
        DB=ZERO;
          DPX<DB;
          DPY<DB

        LDMA;
          DB=5

        DB=ZERO;
          DPX(1)<DB

        MOV M,ICNT

        LDSPI JBI;
          DB=MD

        LDDPA;DB=5

          LDDA;
          DB=5
        LDMA;
          DB=AZ
        NOP
        INCMA
        DPX(-2)<DB;
          DB=MD
        INCMA
        DPY(-3)<DB;
          DB=MD
        DPX(-3)<DB;
          DB=MD
        INCMA
        DPX(-3)<DB;
        DB=MD
        INCMA
        DPY(-4)<DB;
          DB=MD
        NOP
        DPX(-4)<DB;
        DB=MD

        LDMA;
          DB=6

        NOP
        NOP
        LDSPI JFI;
          DB=MD

I2LP:LDDPA;DB=5
      DB=ZERO;                                   "ZERO TROW
          DPY(-2)<DB

        INC JFI
        MOV JFI,JWA

        INC JBI

        LDSPI JCNT;
          DB=5
      SUB M,JWA
        LDTMA;
```

```
          DB=JTMA-1

"SAVE 5 FRONT END VALUES IN TM

        MOV JFI,JRA                           "GET FIRST J IN

        ADD M,JRA;SETMA

        NOP

        ADD M,JRA;
          SETMA                               "GET SECOND J IN
J2LP:INCTMA;

          OUT;
          DB=MD;
          DEC JCNT

        ADD M,JRA;
          SETMA;
          BGT J2LP

        LDSPI JCNT;
          DB=4

"READ 5 FROM BACK NO WRITE

        MOV JBI,JRA;                          "GET J
          SETMA

        NOP

        NOP

        FMUL DPX(-2),MD                       "A5*J

        FMUL DPY(-3),MD                       "A4*J

        FMUL

        FMUL DPX(-3),MD;                      "A3*J
          FADD FM,DPY(3)                      "(A5*J)+S10 (=J1)

        FMUL DPY(-4),MD;                      "A2*J
          FADD FM,DPX(3)                      "(A4*J)+S9 (=S10)

        FADD

        FMUL;
          FADD FM,DPX(2);                     "(A3*J)+S8 (=S9)
          DPY(3)<FA                           "STO S10

        FMUL DPX(-4),MD;                      "A1*J
          FADD FM,DPY(2)                      "(A2*J)+S7 (=S8)

        FMUL DPY(-4),MD;                      "A2*J
          FADD FM,DPX(1);                     "(A1*J)+S6 (=S7)
          DPX(3)<FA                           "STO S9

J3LP:FMUL DPX(-3),MD;                         "A3*J
```

```
         FADD DPY(1),MD;                         "S5+J (=S6)
         DPX(2)<FA;                              "STO S8
         ADD M,JRA;                              "GET NEXT J
         SETMA

       FMUL DPY(-3),MD;                          "A4*J
         FADD FM,DPX(0);                         "(A1*J)+S4 (=S5)
         DPY(2)<FA                               "STO S7

       FMUL DPX(-2),MD;                          "A5*J (=S1)
         FADD FM,DPY(0);                         "(A2*J)+S3 (=S4)
         DPX(1)<FA                               "STO S6

       FMUL DPX(-2),MD;                          "A5*J1
         FADD FM,DPY(-1);                        "(A3*J)+S2 (=S3)
         DPY(1)<FA                               "STO S6

       FMUL DPY(-3),MD;                          "A4*J1
         FADD FM,DPX(-1);                        "(A4*J)+S1 (=S2)
         DPX(0)<FA                               "STO S4

       FMUL;FADD;
         DPX(-1)<FM;                             "STO S1
         DPY(0)<FA                               "STO S3

       FMUL DPX(-3),MD;                          "A3*J1
         FADD FM,DPY(3)                          "(A5*J1)+S10 (=J)

       FMUL DPY(-4),MD;                          "A2*J1
         FADD FM,DPX(3);                         "(A4*J)+S9 (=S10)
         DPY(-1)<FA                              "STO S2

       FMUL DPX(-4),MD;FADD                               "A1*J1

       FMUL;
         FADD FM,DPX(2);                         "(A3*J1)+S8 (=S9)
         DPY(3)<FA                               "STO S10

       FMUL DPX(-4),MD;                          "A1*J1
         FADD FM,DPY(2);                         "(A2*J1)+S7 (=S8)
         DEC JCNT

       FMUL DPY(-4),MD;                          "A2*J1
         FADD FM,DPX(1);                         "(A1*J1)+S6 (=S7)
         DPX(3)<FA;                              "STO S9
         BGT J3LP

"READ 5 FROM FRONT, NO WRITE

       LDSPI JCNT;
         DB=5

       MOV JFI,JRA


J4LP:FMUL DPX(-3),MD;
         FADD DPY(1),MD;
         DPX(2)<FA;
         ADD M,JRA;
         SETMA
```

```
          FMUL DPY(-3),MD;
            FADD FM,DPX(0);
            DPY(2)<FA

          FMUL DPX(-2),MD;
            FADD FM,DPY(0);
            DPX(1)<FA

          FMUL DPX(-2),MD;
            FADD FM,DPY(-1);
            DPY(1)<FA

          FMUL DPY(-3),MD;
            FADD FM,DPX(-1);
            DPX(0)<FA

          FMUL;FADD;
            DPX(-1)<FM;
            DPY(0)<FA

          FMUL DPX(-3),MD;
            FADD FM,DPY(3)

          FMUL DPY(-4),MD;
            FADD FM,DPX(3);
            DPY(-1)<FA

          FMUL DPX(-4),MD;FADD

          FMUL;
            FADD FM,DPX(2);
            DPY(3)<FA

          FMUL DPX(-4),MD;
            FADD FM,DPY(2);
            DEC JCNT

          FMUL DPY(-4),MD;
            FADD FM,DPX(1);
            DPX(3)<FA;
            BGT J4LP

"READ N-5 FROM MIDDLE, WITH WRITE

          LDSPI JCNT;
            DB=NV-5



J5LP:FMUL DPX(-3),MD;
            FADD DPY(1),MD;
            DPX(2)<FA;
            ADD M,JRA;
            SETMA

          FMUL DPY(-3),MD;
            FADD FM,DPX(0);
            DPY(2)<FA

          FMUL DPX(-2),MD;
            FADD FM,DPY(0);
            DPX(1)<FA
```

```
        FMUL DPX(-2),MD;
          FADD FM,DPY(-1);
          DPY(1)<FA

        FMUL DPY(-3),MD;
          FADD FM,DPX(-1);
          DPX(0)<FA

        FMUL;FADD;
          DPX(-1)<FM;
          DPY(0)<FA

        FMUL DPX(-3),MD;
          FADD FM,DPY(3)

        FMUL DPY(-4),MD;
          FADD FM,DPX(3);
          DPY(-1)<FA

        FMUL DPX(-4),MD
          FADD DPY(-2),FA;
          ADD M,JWA;
          SETMA;
          MI<FA

        FMUL;
          FADD FM,DPX(2);
          DPY(3)<FA

        FMUL DPX(-4),MD;
          FADD FM,DPY(2);
        DPY(-2)<FA;
          DEC JCNT

        FMUL DPY(-4),MD;
          FADD FM,DPX(1);
          DPX(3)<FA;
          BGT J5LP

"READ 5 FROM TM STORE,WRITE

        LDTMA;
          DB=JTMA-1
      LDSPI JCNT;DB=5


J6LP:FMUL DPX(-3),MD;
          FADD DPY(1),MD;
          DPX(2)<FA;
        INCTMA

        FMUL DPY(-3),MD;
          FADD FM,DPX(0);
          DPY(2)<FA

        FMUL DPX(-2),MD;
          FADD FM,DPY(0);
          DPX(1)<FA

        FMUL TM,DPX(-2);
          FADD FM,DPY(-1);
          DPY(1)<FA
        FMUL TM,DPY(-3);
          FADD FM,DPX(-1);
          DPX(0)<FA

        FMUL;FADD;
          DPX(-1)<FM;
          DPY(0)<FA

        FMUL TM,DPX(-3);
          FADD FM,DPY(3)

        FMUL TM,DPY(-4);
          FADD FM,DPX(3);
          DPY(-1)<FA

        FMUL TM,DPX(-4);
          FADD DPY(-2),FA;
          ADD M,JWA;
          SETMA;
          MI<FA

        FMUL;
          FADD FM,DPX(2);
          DPY(3)<FA

        FMUL TM,DPX(-4);
          FADD FM,DPY(2);
        DPY(-2)<FA;
          DEC JCNT

        FMUL TM,DPY(-4);
          FADD FM,DPX(1);
          DPX(3)<FA;
          BGT J6LP

"FINISH ROW OPERATIONS

        SUB# ICNT,COSFA;
          SETMA                                          "GET COS

        LDDPA;
          DB=7

        SUB# ICNT,SINFA;
          SETMA                                          "GET SIN

        FMUL DPY(-4),MD

        FADD DPX(3),DPY(-4);
          FMUL

        FMUL DPY(-4),MD;
          FADD

        FADD FM,DPY(2);
          DPX(3)<FA;
          FMUL

        FADD;
          FMUL

        FADD FM,DPX(2);
          DPY(2)<FA
        FADD

        DPX(2)<FA

        DEC ICNT

        BEQ STEP2

        JMP I2LP
STEP2:RETURN
        $END

>
```

```
        $TITLE EXPDO
        $ENTRY EXPDO
        $EXT  NEXTMD    "IN RLNLF - RETURNS ADDR IN MD
    "<SN1> := EXP(Z1*<S1>+Z2*<S2>)
    "VERSION 1-25-78
     "NEW SPAD-TM
      "MV.
    "+SP4 :=1
     "REMOVE SPFTMA

    $EXT VSMUL,VADD,VEXP     "LINK FROM APLIB.FRB
    MV = 16.

    SN1Z = 20.
    "MD ADDRESSES (RELATIVE TO NEXTMD)
    S1Z = 0.
    S2Z = S1Z+MV
    INBUF = S2Z+MV
    Z1Z = INBUF
    Z2Z = Z1Z+1

    T1Z = Z2Z+1
    T2Z = T1Z+MV
    "NEXT FREE = T2Z+MV

    STMADR = 10500K     "START SAVE ADDR IN TM FOR SPAD

EXPDO: NOP
    MOV 0,0; DPX<SPFN
    LDSPI 0;DB=0.
    MOV 0,0;SETMA; MI<DPX   "MD(0):=INITSW
    NOP

    JSR NEXTMD
    LDSPI 1;DB=1
    LDSPI 2;DB=Z1Z
    ADD 0,2   "GET ABSOLUTE ADDR
    LDSPI 3; DB=T1Z
    ADD 0,3
    LDSPI 4; DB=1
    LDSPI 5; DB=MV
    JSR  VSMUL    "<T1> = <S1>*Z1

    JSR NEXTMD
    LDSPI 1;DB=1
    LDSPI 2; DB=Z2Z
    ADD 0,2
    LDSPI 3; DB=T2Z
    ADD 0,3
    LDSPI 4; DB=1
    LDSPI 5; DB=MV
    LDSPI 6;DB=S2Z
    ADD 0,6
    MOV 6,0
    JSR  VSMUL    "<T2> = <S2>*Z2
JSR NEXTMD
LDSPI 1;DB=1
LDSPI 2; DB=T2Z
ADD 0,2
LDSPI 3; DB=1
LDSPI 4; DB=SN1Z
LDSPI 5; DB=1
LDSPI 6; DB=MV
LDSPI 7;DB=T1Z
ADD 0,7
MOV 7,0
JSR   VADD      "<SN1> = <T1>+<T2>

LDSPI 0; DB=SN1Z
LDSPI 1;DB=1
LDSPI 2; DB=SN1Z
LDSPI 3; DB=1
LDSPI 4; DB=MV
JSR  VEXP           "<SN1> = EXP(<SN1>)

LDMA; DB=0.    "RESTORE INITSW FROM MD(0)
NOP
NOP
NOP
LDSPI 0;DB=MD
NOP
RETURN
     $END

>
```

```
DEFINE ME(AANEW,AAOLD,ASC1,AADLT,ASC2,ASC3,AGA,AXP2,ACLF,ANORM)
LOCAL K,KM,B,D
K=16
KM=240
B=AAOLD+15
LOOP:CALL VFILL(B,ASC1,1,16)
CALL VSMUL(ASC1,1,AGA,ASC2,1,16)
CALL VFILL(AADLT,ASC3,1,16)
CALL VADD(ASC3,1,ASC2,1,ASC1,1,16)
CALL VADD(ASC1,1,AANEW,1,ASC2,1,16)
CALL VSQ(ASC2,1,ASC1,1,16)
CALL VSMUL(ASC1,1,AXP2,ASC2,1,16)
CALL VEXP(ASC2,1,ASC3,1,16)
D=ACLF+KM
CALL VSMUL(ASC3,1,ANORM,D,1,16)
B=B-1
KM=KM-16
K=K-1
IF K>0 COTO LOOP
END

>
```

```
          $TITLE TTMOV
          $ENTRY TTMOV,3
          A $EQU 0
          C $EQU 1
          N $EQU 2
TTMOV:MOV A,A;SETMA
          DEC C;SETTMA
          INCMA
          LDDA;
          DB=5
LOOP:INCMA;
          DPX<MD;
          DEC N
          OUT;DB=DPX;INCTMA;
          BNE LOOP
          RETURN
$END

>
```

```
        "W.FSO
        $TITLE W
        $ENTRY W,0
        $EXT STHIRD
"S PAD ADDRESSES
        B $EQU 0
        I $EQU 8.
        J $EQU 9.
        NJ $EQU 10.
W:LDSPI I;
          DB=96.
        LDSPI NJ;
            DB=1908.

LOOPI:LDSPI J;DB=16.
LOOPJ:MOV NJ,B

        JSR STHIRD

        INC NJ

        DEC J

        BGT LOOPJ

        DEC I

        BGT LOOPI

        RETURN
 $END

 >
```

```
        "STHIRD.FSO
        $TITLE STHIRD
        $ENTRY STHIRD,1

        "SIZING
        MV $EQU 16.
        NV $EQU 96.
        MN1V $EQU MV*NV+MV
"TABLE MEMORY ADDRESSES
        ISTAT $EQU 11000
"S PAD ADDRESSES
        ITOPS $EQU 0
        M $EQU 1
        MN1 $EQU 2
        KBIAS $EQU 3
        MKK $EQU 4
        MK $EQU 5
        TMINC $EQU 6
STHIRD:LDSPI KBIAS;
          DB=0

        LDSPI M;
          DB=MV

        LDSPI MK;
          DB=MV
        LDSPI MN1;
        DB=MN1V

        LDSPI TMINC;
          DB=ISTAT

        LDTMA;
          DB=!ZERO

        LDDPA;
          DB=0

        DPX(0)<TM;
          DPY(0)<TM;
          DEC MK

LPDPXY:DPX(0)<TM;
        DPY(0)<TM;
          DEC MK;
          INCDPA;
          BGT LPDPXY

        LDSPI MK;
          DB=MV

        LDTMA;
          DB=ISTAT

LOOPK:ADD# ITOPS,KBIAS;
          SETMA

        LDSPI MKK;
          DB=MV-8.
```

```
       DB=31.
       INCTMA

       FMUL TM,MD;INCTMA

       FMUL TM,MD;
         INCTMA

       FMUL TM,MD;
         INCTMA;
         INCDPA

       FMUL TM,MD;
         INCTMA;
         INCDPA;
         FADD FM,DPX(0)

       FMUL TM,MD;
         INCTMA;
         INCDPA;
         FADD FM,DPX(0)

       FMUL TM,MD;
         INCTMA;
         INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA

       FMUL TM,MD;
         INCTMA;
         INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA;
         DEC MKK

LOOPKK:FMUL TM,MD;
         INCTMA;
         INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA;
         DEC MKK;
         BGT LOOPKK

       FMUL TM,MD;
         INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA

       FMUL;
         INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA

       FMUL;
         INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA

       INCDPA;
         FADD FM,DPX(0);
         DPX(-2)<FA
```

```
          FADD;
            DPX(-2)<FA

          INCDPA;
            DPX(-2)<FA;
            DEC MK

          ADD MN1,KBIAS;
          BEQ STEP
          JMP LOOPK

STEP:LDDPA;
            DB=0
          LDSPI MK;
            DB=MV

          SUB MN1,ITOPS

LOOPR:INCDPA;
          DEC MK

          ADD MN1,ITOPS;
            SETMA;
            MI<DPX(-1);
            BGT LOOPR

          NOP
          RETURN
$END

'>
```

```
DEFINE XSUM(ITOPS,M,AXJ,KBIAS)
LOCAL K,A,B
K=16
LOOP:K=K-1
A=ITOPS+K
B=AXJ+K
CALL SVE(A,M,B,KBIAS)
IF K>0 GOTO LOOP
END
```

>

```
DEFINE ZSUM(ITOPS,M,AZJ,KBIAS)
LOCAL K,A,B,L,C
K=16
L=KBIAS-M
C=15*KBIAS
A=ITOPS+C
LOOP:K=K-1
B=AZJ+K
CALL SVE(A,1,B,L)
A=A-KBIAS
IF K>0 GOTO LOOP
END
```

>

REFERENCES

[1] R.S. Bucy, K.D. Senne "New Frontiers in Nonlinear Filtering" Lincoln Lab., Technical Note, 1978 - 16, Lexington, Mass.

[2] R.S. Bucy, K.D. Senne, H. Youssef, "Pipeline Parallel and Serial Realizations of Phase Demodulators," Institute for Computer Applications in Science and Engineering, ICASE, Report #76-31, Nov. 1976, NASA Langley, Hampton, Virginia.

[3] R.S. Bucy, C. Hecht, K.D. Senne, "New Methods for Nonlinear Filtering," Revue Francais d'Automatique, Informatique, Recherche Operationelle, J-1, 1973, 3-54.

[4] R.S. Bucy, C. Hecht, and K.D. Senne, "An Engineer's Guide to Building Nonlinear Filters," Final Report SRL-TR-72-0004, Project 7904, Frank J. Seiler Research Laboratory, USAF Academy, Colorado, (1972), DDC-AD-746921/2.

(19) AFOSR

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR-79-0923 USCAE 137 (19)	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) SOFTWARE FOR NONLINEAR FILTERING. (14)		5. TYPE OF REPORT & PERIOD COVERED Interim Scientific rept., USCAE-
		6. PERFORMING ORG. REPORT NUMBER USCAE-53-4514-1787, 53-4514-1792
7. AUTHOR(s) (10) R.S. Bucy, F. Ghovanlou A.J. Mallinckrodt, K.D. Senne		8. CONTRACT OR GRANT NUMBER(s) (15) F44620-76-C-0085 (Contract) AFOSR-76-3100 (Grant)
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Southern California Department of Aerospace Engineering Los Angeles, California 90007		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (17) (16) 2305 B1 2304 A1
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Electronic and Solid State Sciences Division AFOSR/NA, Bolling AFB, Washington, D.C. 20332		12. REPORT DATE (11) June 1979
		13. NUMBER OF PAGES 129
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 135p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

"The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research of the U.S. Government."

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

"This document has been approved for public release and sale; its distribution is unlimited."

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software	7600	Demodulation
Nonlinear Filtering	6600	
Star 100	Illiac	
AP 120B	Phase Lock Loop	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

As part of a continuing search for the ideal architecture for performing the computations required to realize a non-linear filter, we have developed software for various machines over the past ten years. A description of the latest software is given in [1], while [2], [3], and [4] are useful for for background information on the non-linear filtering problem as well as comments about software efficiencies relevant to various machines.

(continued)

We started our studies over 10 years ago using the CDC 6600 at the Aerospace Corporation and Kirkland AFB, and continuing at Eglin AFB, see [4]. At the Institute for Advanced Computation, we gained access to the Illiac IV and at ICASE, Nasa Langley, the Star 100, see [2]. Access to the Cray was obtained through Cray Research and later NCAR. Experiments on the AP120B array processor were possible because of the acquisition of one here at USC used in conjunction with a PDP 11-55.

The purpose of this report is to document the current software, for all these machines. In particular, we have found [2], with the listings of the 6600 and Star Codes, extremely useful in the past, although now these listings are outdated. In particular, the assembly language coding for the AP-120B involved extensive effort over a long time period and should be documented so that others interested in similar problems, can avoid the pain of developing the software from scratch.