```
C PROGRAM TO FIT A TWO-PEAK MODEL IN WHICH ONE DISTRIBUTION OF PINNING
C SITE STRENGTHS IS LOG-NORMAL AND THE OTHER IS NORMAL.
C INPUT IS A DATA FILE WHOSE LOCATION IS SPECIFIED IN A TEMPLATE FILE.
C OUTPUTS ARE:    1. A FILE CONTAINING REAL AND FITTED DATA,
C                 2. A LOGFILE WITH DETAILS OF PROGRESS AND ERRORS
C                 3. A GNUPLOT SCRIPT FILE TO AUTOMATICALLY PLOT (1.)

      IMPLICIT NONE
      INTEGER I, M
      REAL S(1000), Y(1000)
      REAL S1, S2
      REAL A1, D1
      REAL A2, D2
      REAL AVSUM
      INTEGER NLINES, IMAX, LIM1, LIM2, IMIN
      REAL SYMAX, YMAX, SYMAX1
      REAL NB
      REAL BL
      INTEGER IBL
      CHARACTER LOC*21
      CHARACTER F*14
      CHARACTER D*14
      CHARACTER SU*14
      INTEGER L1, L2, L3

      LOC = 'C:/Data/Barkhausen02/'

C CREATE A FILE TO DUMP TEMPORARY DATA

      OPEN(3, FILE =LOC//'temp')
      CALL OPEN(F, D, SU, L1, L2, L3)

C READ IN REAL DATA POINTS

      NLINES = 0
      DO 1 I = 1, 1000
         READ(2, *, IOSTAT = M) S(I), Y(I)
         IF (M .LT. 0) THEN
            GOTO 6
         ELSE IF (M .GT. 0) THEN
            WRITE (6,*) 'ERROR IN OPENING FILE'
            WRITE (8,*) 'ERROR IN OPENING FILE'
            GOTO 6
         ENDIF
         NLINES = NLINES + 1
    1 CONTINUE
    6 REWIND(2)

C FIND THE VALUE OF THE NOISE BASELINE

      CALL MAXY(S, Y, 0, NLINES, SYMAX, YMAX, IMAX)
      CALL NOISE(S, Y, NLINES, YMAX, NB)
      CALL BASEL(S, Y, NLINES, BL, IBL, NB)

C OBTAIN STARTING VALUES OF S1 AND S2

      S1 = 1.0
      S2 = 0.0
      CALL STARTS(S, Y, NLINES, SYMAX1, S2, IMIN)
      S1 = LOG(SYMAX1 - BL)

C OBTAIN STARTING VALUES OF A1 AND D1

      IF (IMIN .EQ. 0) THEN
         LIM1 = IBL
         LIM2 = NLINES
         A1 = 0.1
         D1 = 0.1
```

```
         CALL ADLOG(S, Y, A1, D1, S1, BL, AVSUM, LIM1, LIM2, NLINES, NB)

C GIVE SECOND PEAK A SLIGHTLY DIFFERENT STARTING VALUE

         S2 = SYMAX1 + (SYMAX1/100)
         A2 = A1 + (A1/100)
         D2 = D1 + (D1/100)
      ELSE

C FIRST MAXIMUM: FIT PEAK

         LIM1 = IBL
         LIM2 = IMIN
         A1 = 0.01
         D1 = 0.1
         CALL ADLOG (S,Y,A1,D1,S1,BL,AVSUM,LIM1,LIM2,NLINES,NB)

C SECOND MAXIMUM: FIT PEAK

         LIM1 = IMIN
         LIM2 = NLINES
         A2 = 0.01
         D2 = 0.1
         CALL ADLIN(S, Y, A2, D2, S2, AVSUM, LIM1, LIM2, NLINES, NB)
      ENDIF

C WRITE DOWN STARTING VALUES

      WRITE (6,*) 'STARTING VALUES'
      WRITE (6,*) 'BL = ', BL
      WRITE (6,*) 'S1 = ', S1
      WRITE (6,*) 'S2 = ', S2
      WRITE (6,*) 'A1 = ', A1
      WRITE (6,*) 'A2 = ', A2
      WRITE (6,*) 'D1 = ', D1
      WRITE (6,*) 'D2 = ', D2
      WRITE (8,*) 'STARTING VALUES'
      WRITE (8,*) 'BL = ', BL
      WRITE (8,*) 'S1 = ', S1
      WRITE (8,*) 'S2 = ', S2
      WRITE (8,*) 'A1 = ', A1
      WRITE (8,*) 'A2 = ', A2
      WRITE (8,*) 'D1 = ', D1
      WRITE (8,*) 'D2 = ', D2

C FIT THE TWO-PEAK MODEL USING ITERATION METHOD

      CALL TWOPEAK(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      CALL GNU(F, D, SU, L1, L2, L3, A1, A2, D1, D2, S1, S2, BL)
      END

      SUBROUTINE BASEL(S, Y, NLINES, BL, IBL, NB)

C FIND THE BASELINE OF THE FIRST (LOG) PEAK BY USING A BIGGER-THAN-NOISE
C CRITERION OF 1.25

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL NB, BL
      INTEGER I, NLINES, IBL

      DO 1 I = 1, NLINES
         IF (Y(I) .GT. (NB+0.01)) THEN
            BL = S(I)
            IBL = I
            WRITE(6,*) 'BASELINE FOUND AT ', BL
            WRITE(8,*) 'BASELINE FOUND AT ', BL
            GOTO 2
```

```
         ENDIF
    1 CONTINUE
    2 RETURN
      END

      SUBROUTINE OPEN(F, D, S, L1, L2, L3)

      IMPLICIT NONE
      CHARACTER LOC*21
      CHARACTER F*14
      CHARACTER D*14
      CHARACTER S*14
      INTEGER L1, L2, L3

      LOC = 'C:/Data/Barkhausen02/'

C READ THE LOCATION OF THE DATA FILE FROM A TEMPLATE

      OPEN(4, FILE =LOC//'Programs/template', STATUS = 'OLD')
      READ(4,*)
      READ(4,*)
      READ(4,9) D
      CALL SHORTEN (D,L1)
      READ(4,9) S
      CALL SHORTEN (S,L2)
      READ(4,9) F
      CALL SHORTEN (F, L3)
      REWIND(4)
      CLOSE(4)
    9 FORMAT(A14)

C MAKE FILE TO OUTPUT REAL DATA AND MODEL FIT
      OPEN(1, FILE = LOC//D(1:L1)//'/'//S(1:L2)//'/'//F(1:L3)//
     &'_logfit')
C OPEN REAL DATA FILE
      OPEN(2, FILE = LOC//D(1:L1)//'/'//S(1:L2)//'/'//F(1:L3),
     & STATUS = 'OLD')
C MAKE LOGFILE TO OUTPUT PROGRESS, FITTING PARAMETERS AND ERRORS
      OPEN(8, FILE = LOC//D(1:L1)//'/'//S(1:L2)//'/'//F(1:L3)//
     &'_log.log')
C MAKE GNUPLOT SCRIPT TO PLOT GRAPH WITH FITTING PARAMETERS
      OPEN(7, FILE=LOC//'/Plotfile/logfit'//F(1:L3)//'.gnu')
      WRITE (6,*) 'ANALYSING DATA FROM FILE'//LOC//D(1:L1)//'/'
     &//S(1:L2)//'/'//F(1:L3)//' USING LOG-LINEAR MODEL'
      WRITE (8,*) 'ANALYSING DATA FROM FILE'//LOC//D(1:L1)//'/'
     &//S(1:L2)//'/'//F(1:L3)//' USING LOG-LINEAR MODEL'
      END

      SUBROUTINE SHORTEN(NAME, LEN)

C TO FIND LENGTH OF CHARACTER STRING AND REMOVE BLANK SPACES IN
C FILENAMES

      IMPLICIT NONE
      CHARACTER NAME*14
      CHARACTER*1 A(14)
      CHARACTER*1 BLANK
      PARAMETER(BLANK = ' ')
      INTEGER N, LEN

      DO 1 N = 1, 14
         A(N) = NAME(N:N+1)
         IF (A(N) .EQ. BLANK) THEN
            GOTO 1
         ELSE
            LEN = N
         ENDIF
    1 CONTINUE
```

```
      RETURN
      END

      SUBROUTINE GNU(F, D, S, L1, L2, L3, A1, A2, D1, D2, S1, S2, BL)

C TO WRITE A GNUPLOT SCRIPT TO DISPLAY REAL DATA, MODEL AND FITTING
C PARAMTERS ON SCREEN AND IN MONO AND COLOUR .EPS FILES

      CHARACTER LOC*21
      CHARACTER F*14
      CHARACTER D*14
      CHARACTER S*14
      INTEGER L1, L2, L3
      REAL A1, A2, D1, D2, S1, S2, BL

      LOC = 'C:/Data/Barkhausen02/'

      WRITE(7,*) '# Instructions to Gnuplot - to plot model and'
     &//' real data'
      WRITE(7,*) 'set autoscale'
      WRITE(7,*) 'set nologscale'
      WRITE(7,*) 'set nogrid'
      WRITE(7,*) 'set nolabel'
      WRITE(7,*) 'set key top right Left'
      WRITE(7,*) 'set title "Log-linear two-point model fit for'
     &//LOC//D(1:L1)//'/'//S(1:L2)//'/'//F(1:L3)//'"'
      WRITE(7,*) 'set xlabel "Magnetising current/A "'
      WRITE(7,*) 'set ylabel "RMS Barkhausen voltage/V"'
      WRITE(7,*) 'set label 1 "Model parameters" at graph 0.02, 0.95'
      WRITE(7,*) 'set label 2 "A_1 = ', A1,'" at graph 0.02, 0.90'
      WRITE(7,*) 'set label 3 "A_2 = ', A2,'" at graph 0.02, 0.85'
      WRITE(7,*) 'set label 4 "{/Symbol=14 D}S_1 = ', D1,'" at'
     &//' graph 0.02, 0.80'
      WRITE(7,*) 'set label 5 "{/Symbol=14 D}S_2 = ', D2,'" at'
     &//' graph 0.02, 0.75'
      WRITE(7,*) 'set label 6 "<S>_1 = ', S1,'" at graph 0.02, 0.70'
      WRITE(7,*) 'set label 7 "<S>_2 = ', S2,'" at graph 0.02, 0.65'
      WRITE(7,*) 'set label 8 "b = ', BL,'" at graph 0.02, 0.60'
      WRITE(7,*) 'set xrange [-0.7:0.7]'
      WRITE(7,*) 'set terminal x11'
      WRITE(7,*) 'plot "'//LOC//D(1:L1)//'/'//S(1:L2)//'/'
     &//F(1:L3)//'_logfit" using 1:2 title "Real data" with lines 1'
      WRITE(7,*) 'replot "'//LOC//D(1:L1)//'/'//S(1:L2)//'/'
     &//F(1:L3)//'_logfit" using 1:3 title "Model" with lines 3'
      WRITE(7,*) 'set output  "'//LOC//D(1:L1)//'/'//S(1:L2)//'/'
     &//F(1:L3)//'_log_mono.ps"'
      WRITE(7,*) 'set terminal postscript eps enhanced mono dashed'
     &//'"Helvetica" 20'
      WRITE(7,*) 'replot'
      WRITE(7,*) 'set output  "'//LOC//D(1:L1)//'/'//S(1:L2)//'/'
     &//F(1:L3)//'_log_colour.ps"'
      WRITE(7,*) 'set terminal postscript eps enhanced color solid'
     &//'"Helvetica" 20'
      WRITE(7,*) 'replot'
      WRITE(7,*) 'set terminal x11'
      WRITE(7,*) 'replot'
      WRITE(7,*)
      WRITE(7,*)
      RETURN
      END

      SUBROUTINE MAXY(S, Y, LIM1, LIM2, SYMAX, YMAX, IMAX)

C FINDS THE LARGEST VALUE OF Y IN THE DATA SET

      IMPLICIT NONE
      INTEGER I, LIM1, LIM2, IMAX
      REAL Y(1000), S(1000), SYMAX, YMAX
```

```
      YMAX = 0.0
      SYMAX = 0.0
      IMAX = 0

      DO 1 I = LIM1, LIM2
         IF (Y(I) .GT. YMAX) THEN
            YMAX = Y(I)
            SYMAX = S(I)
            IMAX = I
         ENDIF
    1 CONTINUE
      RETURN
      END

      SUBROUTINE MINY(S, Y, LIM1, LIM2, SYMIN, YMIN, IMIN)

C FINDS THE SMALLEST VALUE OF Y IN THE DATA SET

      IMPLICIT NONE
      INTEGER I, LIM1, LIM2, IMIN
      REAL Y(1000), S(1000), SYMIN, YMIN

      YMIN = Y(LIM1)
      SYMIN = 0.0
      IMIN = 0

      DO 1 I = LIM1, LIM2
         IF (Y(I) .LT. YMIN) THEN
            YMIN = Y(I)
            SYMIN = S(I)
            IMIN = I
         ENDIF
    1 CONTINUE
      RETURN
      END

      SUBROUTINE BIN(S, Y, YMAX, NLINES, COUNT, BINA)

C DIVIDES DATA INTO 'BINS' ACCORDING TO SIZE

      IMPLICIT NONE
      INTEGER I, J, NLINES, COUNT(1000)
      REAL Y(1000), S(1000), YMAX, BINA, BINU, BINL

      BINA = YMAX/REAL(NLINES)
      BINL = 0.0
      BINU = 0.0

      DO 1 I = 1, NLINES
         BINL = BINU
         BINU = REAL(I)*BINA
         COUNT(I) = 0
         DO 2 J = 1, NLINES
            IF ((Y(J) .GE. BINL) .AND. (Y(J) .LT. BINU)) THEN
               COUNT(I) = COUNT(I) + 1
            ENDIF
    2    CONTINUE
    1 CONTINUE
      END

      SUBROUTINE NOISE(S, Y, NLINES, YMAX, NB)

C FINDS THE VALUE OF THE NOISE BASELINE AND LIMITS AROUND IT.

      IMPLICIT NONE
      REAL S(1000), Y(1000), YMAX, BINA
      INTEGER I, NLINES, COUNT(1000), MAXC, IMAXC
```

```
      REAL NB
      REAL BBINUP, BBINDO, NOISUM
      INTEGER INC, J

      NB = 0.0
      MAXC = 0
      IMAXC = 0

C DIVIDE DATA INTO BINS
      CALL BIN(S, Y, YMAX, NLINES, COUNT, BINA)
C FIND BIN WITH LARGEST NUMBER OF DATA POINTS IN IT
C THIS IS THE 'NOISE BASELINE' BIN
      DO 1 I = 1, NLINES
         IF (COUNT(I) .GT. MAXC) THEN
            MAXC = COUNT(I)
            IMAXC = I
         ENDIF
    1   CONTINUE

C FIND AVERAGE OF ALL POINTS THAT ARE IN THIS BIN OR OTHERS AROUND IT
C CHANGE VALUE '3' TO ANOTHER INTEGER TO CHANGE SENSITIVITY.

      BBINUP = REAL(IMAXC + 3)*BINA
      BBINDO = REAL(IMAXC - 3)*BINA

      INC = 0

      DO 2 J = 1, NLINES
         IF ((Y(J) .GE. BBINDO) .AND. (Y(J) .LT. BBINUP)) THEN
            NOISUM = NOISUM + Y(J)
            INC = INC + 1
         ENDIF
    2 CONTINUE

      NB = NOISUM/REAL(INC)
      WRITE (6,*) 'NOISE = ', NB
      WRITE (8,*) 'NOISE = ', NB
      RETURN
      END

      SUBROUTINE MAXY2(S, Y, NLINES, SYMAX1, SYMAX2, YMAX1, YMAX2, IMIN)

C TO FIND THE POSITIONS OF THE PEAK(S) AND MINIMUM IF IT EXISTS

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL AVES(20), AVEY(20), SSUM, YSUM
      REAL AVEM1, AVEM2, AVEM3
      REAL SYMAX1, SYMAX2, SYMIN
      REAL YMAX1, YMAX2, YMIN
      INTEGER NLINES, COUNT, I, J, POINTS
      INTEGER JM1, JM2, JM3, TEMP
      INTEGER IMAX1, IMAX2, IMIN
      INTEGER LIM1, LIM2

C DIVIDE THE POINTS INTO TWENTY GROUPS AND FIND THE AVERAGE.

      POINTS = NLINES/20

      DO 1 J = 1, 20
         COUNT = 0
         YSUM = 0.0
         SSUM = 0.0
         DO 2 I = (J-1)*POINTS, J*POINTS
            YSUM = YSUM + Y(I)
            SSUM = SSUM + S(I)
            COUNT = COUNT+1
    2    CONTINUE
```

```
         AVEY(J) = YSUM/REAL(COUNT)
         AVES(J) = SSUM/REAL(COUNT)
         WRITE(3,*) J, AVES(J), AVEY(J)
    1 CONTINUE

      AVEM1 = 0.0
      AVEM2 = 0.0
      AVEM3 = 0.0
      JM1 = 0
      JM2 = 0
      JM3 = 0

C FIND THE TWO LARGEST VALUES OF THE AVERAGE

      DO 3 J = 1, 20
         IF (AVEY(J) .GT. AVEM1) THEN
            AVEM2 = AVEM1
            JM2 = JM1
            AVEM1 = AVEY(J)
            JM1 = J
         ELSE IF ((AVEY(J) .GT. AVEM2) .AND. (AVEY(J) .LE. AVEM1)) THEN
            AVEM2 = AVEY(J)
            JM2 = J
         ENDIF
    3 CONTINUE
      WRITE (6,*) 'LARGEST TWO VALUES ARE AT ', JM1, JM2
      WRITE (8,*) 'LARGEST TWO VALUES ARE AT ', JM1, JM2


C ARE THE TWO MAXIMA NEXT TO EACH OTHER?

      IF (ABS(JM1 - JM2) .NE. 1) THEN

C IF NOT, THEN WE CAN EASILY FIND THE PEAKS AND THE MINIMUM BETWEEN THEM
C MAKE JM1 THE SMALLER OF THE TWO J-VALUES

         IF (JM1 .GT. JM2) THEN
            TEMP = JM1
            JM1 = JM2
            JM2 = TEMP
         ENDIF

C FIND THE MAXIMUM NEAR J1 PRECISELY

         LIM1 = (JM1-1)*POINTS
         LIM2 = (JM1+1)*POINTS

         CALL MAXY(S, Y, LIM1, LIM2, SYMAX1, YMAX1, IMAX1)

C FIND THE MINIMUM NEAR J2 PRECISELY

         LIM1 = (JM2-1)*POINTS
         LIM2 = (JM2+1)*POINTS
         CALL MAXY(S, Y, LIM1, LIM2, SYMAX2, YMAX2, IMAX2)

C FIND THE MINIMUM SITUATED BETWEEN THESE TWO MAXIMA

         LIM1 = IMAX1
         LIM2 = IMAX2
         CALL MINY(S, Y, LIM1, LIM2, SYMIN, YMIN, IMIN)
         WRITE(6,*) 'MAXIMUM 1 AT ', SYMAX1, ' HEIGHT ', YMAX1
         WRITE(6,*) 'MINIMUM AT ', SYMIN, ' HEIGHT ', YMIN
         WRITE(6,*) 'MAXIMUM 2 AT ', SYMAX2, ' HEIGHT ', YMAX2
         WRITE(8,*) 'MAXIMUM 1 AT ', SYMAX1, ' HEIGHT ', YMAX1
         WRITE(8,*) 'MINIMUM AT ', SYMIN, ' HEIGHT ', YMIN
         WRITE(8,*) 'MAXIMUM 2 AT ', SYMAX2, ' HEIGHT ', YMAX2
      ELSE
```

```
C IF THE TWO MAX AVERAGE VALUES ARE NEXT TO ONE ANOTHER
C TRY AND FIND ANOTHER PEAK IN THE AVERAGE VALUES

      DO 4 J = 2, 19

C IS J THE POSITION OF A PEAK?

        IF ((AVEY(J) .GT. AVEM3) .AND. (AVEY(J) .GE. AVEY(J+1))) THEN

C IS AVEY(J) LESS THAN THE MAX PEAK VALUE? AND IS IT GREATER THAN A
C THRESHOLD VALUE (ARBITRARILY GIVEN AS AVEM1/10 TO ELIMINATE NOISE)?

          IF ((AVEY(J) .LT. AVEM1) .AND. (AVEY(J) .GT. (AVEM1/10)))
     &      THEN

C IF THE NEW J-VALUE IS MORE THAN 1 UNIT AWAY FROM JM1
C THIS SHOULD ONLY FAIL IF THERE ARE 3 PEAKS OF EXACT SAME HEIGHT NEXT
C TO ONE ANOTHER

            IF (ABS(JM1 - J) .GT. 1) THEN

C DEFINE NEW AVERAGE MAX VALUE AND J VALUE

              AVEM3 = AVEY(J)
              JM3 = J
            ENDIF
          ENDIF
        ENDIF
    4 CONTINUE

C IF WE HAVE OBTAINED A NON-ZERO VALUE OF JM3 FROM THIS PROCEDURE THEN

      IF (JM3 .NE. 0) THEN

C REDEFINE IT AS JM2

        JM2 = JM3

C MAKE JM1 SMALLER THAN JM2

        IF (JM1 .GT. JM2) THEN
          TEMP = JM1
          JM1 = JM2
          JM2 = TEMP
        ENDIF

C FIND PRECISE POSITION OF MAX NEAR JM1

        LIM1 = (JM1-1)*POINTS
        LIM2 = (JM1+1)*POINTS
        CALL MAXY(S, Y, LIM1, LIM2, SYMAX1, YMAX1, IMAX1)

C FIND PRECISE POSITION OF MAX NEAR JM2

        LIM1 = (JM2-1)*POINTS
        LIM2 = (JM2+1)*POINTS
        CALL MAXY(S, Y, LIM1, LIM2, SYMAX2, YMAX2, IMAX2)

C FIND PRECISE POSITION OF MINIMUM IN BETWEEN THESE

        LIM1 = IMAX1
        LIM2 = IMAX2
        CALL MINY(S, Y, LIM1, LIM2, SYMIN, YMIN, IMIN)
        WRITE(6,*) 'MAXIMUM 1 AT ', SYMAX1, ' HEIGHT ', YMAX1
        WRITE(6,*) 'MINIMUM AT ', SYMIN, ' HEIGHT ', YMIN
        WRITE(6,*) 'MAXIMUM 2 AT ', SYMAX2, ' HEIGHT ', YMAX2
        WRITE(8,*) 'MAXIMUM 1 AT ', SYMAX1, ' HEIGHT ', YMAX1
        WRITE(8,*) 'MINIMUM AT ', SYMIN, ' HEIGHT ', YMIN
```

```
            WRITE(8,*) 'MAXIMUM 2 AT ', SYMAX2, ' HEIGHT ', YMAX2
         ELSE

C IF THE TWO MAXIMA ARE NEXT TO ONE ANOTHER AND THERE ARE NO OTHER
C PEAKS: MAKE JM1 SMALLER THAN JM2

            IF (JM1 .GT. JM2) THEN
               TEMP = JM1
               JM1 = JM2
               JM2 = TEMP
            ENDIF

C FIND MAXIMUM NEAR JM1 AND JM2

            LIM1 = (JM1-1)*POINTS
            LIM2 = (JM2+1)*POINTS
            CALL MAXY(S, Y, LIM1, LIM2, SYMAX1, YMAX1, IMAX1)

C PUT BOTH S-VALUES AT THIS POINT.

            SYMAX2 = SYMAX1
            YMAX2 = YMAX1
            IMAX2 = IMAX1
            IMIN = 0
            WRITE(6,*) 'SINGLE MAXIMUM AT ', SYMAX1, ' HEIGHT ', YMAX1
            WRITE(6,*) 'NO MINIMUM'
            WRITE(8,*) 'SINGLE MAXIMUM AT ', SYMAX1, ' HEIGHT ', YMAX1
            WRITE(8,*) 'NO MINIMUM'
         ENDIF
      ENDIF
      RETURN
      END

      SUBROUTINE TWOPEAK(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIT A TWO-PEAK MODEL BY ITERATING USING STARTING VALUES ALREADY
C OBTAINED.
C FORCED TO ITERATE AT LEAST 10 TIMES THROUGH WHOLE PROCEDURE

      IMPLICIT NONE
      REAL S(1000), Y(1000), A1, A2, D1, D2, S1, S2, AVSUM, NB, BL
      INTEGER NLINES, ITER
      REAL ERROR, OLDERROR

    9 WRITE(6,*) 'USING BEGINNING VALUES, ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'USING BEGINNING VALUES, ERROR = ',(AVSUM*100),' %'
      CALL BESTA1(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW A1 = ', A1, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW A1 = ', A1, ' ERROR = ',(AVSUM*100),' %'
      CALL BESTA2(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW A2 = ', A2, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW A2 = ', A2, ' ERROR = ',(AVSUM*100),' %'
      CALL BESTD1(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW D1 = ', D1, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW D1 = ', D1, ' ERROR = ',(AVSUM*100),' %'
      CALL BESTD2(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW D2 = ', D2, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW D2 = ', D2, ' ERROR = ',(AVSUM*100),' %'
      CALL BESTS1(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW S1 = ', S1, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW S1 = ', S1, ' ERROR = ',(AVSUM*100),' %'
      CALL BESTS2(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW S2 = ', S2, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW S2 = ', S2, ' ERROR = ',(AVSUM*100),' %'
      CALL BESTBL(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      WRITE(6,*) 'NEW BL = ', BL, ' ERROR = ',(AVSUM*100),' %'
      WRITE(8,*) 'NEW BL = ', BL, ' ERROR = ',(AVSUM*100),' %'
```

```
      OLDERROR = ERROR
      ERROR = AVSUM

      IF ((ERROR .LT. OLDERROR) .OR. (ITER .LE. 10)) THEN
         ITER = ITER + 1
         WRITE (6,*) 'OVERALL ITERATION NO. ', ITER
         WRITE (6,*) '*************************'
         WRITE (8,*) 'OVERALL ITERATION NO. ', ITER
         WRITE (8,*) '*************************'
         GOTO 9
      ENDIF
      RETURN
      END

      SUBROUTINE ADLOG(S,Y,A,D,SV,BL,AVSUM,LIM1,LIM2,NLINES,NB)

C TO OBTAIN STARTING VALUES OF A AND D GIVEN A VALUE OF S
C ONLY FITS WITHIN LIMIT SPECIFIED
C FORCED TO ITERATE AT LEAST 10 TIMES TO PREVENT SPURIOUS MINIMA

      IMPLICIT NONE
      INTEGER ITER
      REAL S(1000), Y(1000), A, D, SV, AVSUM, BL
      INTEGER NLINES, LIM1, LIM2
      REAL ERROR, NB
      REAL OLDERROR

      ITER = 0

      WRITE (6,*) 'CALCULATING STARTING VALUES'
      WRITE (8,*) 'CALCULATING STARTING VALUES'

    5 CALL ALOG(S, Y, A, D, SV, BL, AVSUM, LIM1, LIM2, NLINES, NB)
      CALL DLOG(S, Y, A, D, SV, BL, AVSUM, LIM1, LIM2, NLINES, NB)
      OLDERROR = ERROR
      ERROR = AVSUM
      IF ((ERROR .LT. OLDERROR) .OR. (ITER .LE. 10)) THEN
         ITER = ITER + 1
         GOTO 5
      ENDIF
      RETURN
      END

      SUBROUTINE ADLIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)

C TO OBTAIN STARTING VALUES OF A AND D GIVEN A VALUE OF S
C WITHIN LIMITS SPECIFIED
C FORCED TO ITERATE AT LEAST 10 TIMES TO PREVENT SPURIOUS MINIMA

      IMPLICIT NONE
      INTEGER ITER
      REAL S(1000), Y(1000), A, D, SV, AVSUM
      INTEGER NLINES, LIM1, LIM2
      REAL ERROR, NB
      REAL OLDERROR

      ITER = 0

      WRITE (6,*) 'CALCULATING STARTING VALUES'
      WRITE (8,*) 'CALCULATING STARTING VALUES'
    5 CALL ALIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)
      CALL DLIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)
      OLDERROR = ERROR
      ERROR = AVSUM
      IF ((ERROR .LT. OLDERROR) .OR. (ITER .LE. 10)) THEN
         ITER = ITER + 1
      ENDIF
      RETURN
```

```
      END

      SUBROUTINE STARTS(S, Y, NLINES, SYMAX1, SYMAX2, IMIN)

C SUBROUTINE TO FIND STARTING VALUES OF S1 AND S2

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL SYMAX1, SYMAX2, YMAX1, YMAX2, TEMP
      INTEGER NLINES, IMIN

      WRITE (6,*) 'AUTOMATIC OPERATION'
      WRITE (8,*) 'AUTOMATIC OPERATION'
      CALL MAXY2(S, Y, NLINES, SYMAX1, SYMAX2, YMAX1, YMAX2, IMIN)
      IF (SYMAX2 .GT. SYMAX1) THEN
      ELSE
          TEMP = SYMAX2
          SYMAX2 = SYMAX1
          SYMAX1 = TEMP
      ENDIF
      WRITE (6,*) 'PEAK 1 AT ', SYMAX1, ' HEIGHT ', YMAX1
      WRITE (6,*) 'PEAK 2 AT ', SYMAX2, ' HEIGHT ', YMAX2
      WRITE (8,*) 'PEAK 1 AT ', SYMAX1, ' HEIGHT ', YMAX1
      WRITE (8,*) 'PEAK 2 AT ', SYMAX2, ' HEIGHT ', YMAX2
      RETURN
      END

      SUBROUTINE BESTA1(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIND THE BEST-FIT VALUE OF A1

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL A1U, A1D, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)
      A1U = A1 + (A1/100)
      CALL FITTWO(S,Y,A1U,A2,D1,D2,S1,S2,BL,AVSUMU,NLINES,NB)
      A1D = A1 - (A1/100)
      CALL FITTWO(S,Y,A1D,A2,D1,D2,S1,S2,BL,AVSUMD,NLINES,NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            A1 = A1D
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            A1 = A1U
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'A1: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'A1: DETECTED MINIMUM ', ITER, ' ITERATIONS'
         ENDIF
      ELSE
            WRITE (6,*) 'A1: TOO MANY ITERATIONS; PROCESS FAILED'
            WRITE (8,*) 'A1: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE BESTA2(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIND THE BEST-FIT VALUE OF A2
```

```
      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL A2U, A2D, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)
      A2U = A2 + (A2/100)
      CALL FITTWO(S,Y,A1,A2U,D1,D2,S1,S2,BL,AVSUMU,NLINES,NB)
      A2D = A2 - (A2/100)
      CALL FITTWO(S,Y,A1,A2D,D1,D2,S1,S2,BL,AVSUMD,NLINES,NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            A2 = A2D
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            A2 = A2U
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'A2: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'A2: DETECTED MINIMUM ', ITER, ' ITERATIONS'
         ENDIF
      ELSE
         WRITE (6,*) 'A2: TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'A2: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE BESTD1(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIND THE BEST-FIT VALUE OF D1

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL D1U, D1D, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)
      D1U = D1 + (D1/100)
      CALL FITTWO(S,Y,A1,A2,D1U,D2,S1,S2,BL,AVSUMU,NLINES,NB)
      D1D = D1 - (D1/100)
      CALL FITTWO(S,Y,A1,A2,D1D,D2,S1,S2,BL,AVSUMD,NLINES,NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            D1 = D1D
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            D1 = D1U
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'D1: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'D1: DETECTED MINIMUM ', ITER, ' ITERATIONS'
         ENDIF
      ELSE
         WRITE (6,*) 'D1: TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'D1: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
```

```
      END

      SUBROUTINE BESTD2(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIND THE BEST-FIT VALUE OF D2

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL D2U, D2D, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)
      D2U = D2 + (D2/100)
      CALL FITTWO(S,Y,A1,A2,D1,D2U,S1,S2,BL,AVSUMU,NLINES,NB)
      D2D = D2 - (D2/100)
      CALL FITTWO(S,Y,A1,A2,D1,D2D,S1,S2,BL,AVSUMD,NLINES,NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            D2 = D2D
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            D2 = D2U
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'D2: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'D2: DETECTED MINIMUM ', ITER, ' ITERATIONS'
         ENDIF
      ELSE
         WRITE (6,*) 'D2: TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'D2: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE BESTS1(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIND THE BEST-FIT VALUE OF S1

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL S1U, S1D, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)
      S1U = S1 + (S1/100)
      CALL FITTWO(S, Y, A1, A2, D1, D2, S1U, S2, BL, AVSUMU, NLINES, NB)
      S1D = S1 - (S1/100)
      CALL FITTWO(S, Y, A1, A2, D1, D2, S1D, S2, BL, AVSUMD, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            S1 = S1D
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            S1 = S1U
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'S1: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'S1: DETECTED MINIMUM ', ITER, ' ITERATIONS'
```

```
         ENDIF
      ELSE
         WRITE (6,*) 'S1: TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'S1: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE BESTS2(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO FIND THE BEST-FIT VALUE OF S2

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL S2U, S2D, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      S2U = S2 + (S2/100)
      CALL FITTWO(S, Y, A1, A2, D1, D2, S1, S2U, BL, AVSUMU, NLINES, NB)
      S2D = S2 - (S2/100)
      CALL FITTWO(S, Y, A1, A2, D1, D2, S1, S2D, BL, AVSUMD, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            S2 = S2D
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            S2 = S2U
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'S2: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'S2: DETECTED MINIMUM ', ITER, ' ITERATIONS'
         ENDIF
      ELSE
         WRITE (6,*) 'S2: TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'S2: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE BESTBL(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO CALCULATE THE BEST-FIT VALUE OF BL
      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A1, A2, D1, D2, S1, S2
      INTEGER NLINES, ITER
      REAL BLU, BLD, AVSUMU, AVSUMD, AVSUM, NB, BL

      ITER = 0

    4 CALL FITTWO(S, Y, A1, A2, D1, D2, S1, S2, BL, AVSUM, NLINES, NB)
      BLU = BL + (BL/100)
      CALL FITTWO(S, Y, A1, A2, D1, D2, S1, S2, BLU, AVSUMU, NLINES, NB)
      BLD = BL - (BL/100)
      CALL FITTWO(S, Y, A1, A2, D1, D2, S1, S2, BLD, AVSUMD, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            BL = BLD
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            BL = BLU
```

```
            ITER = ITER + 1
            GOTO 4
         ELSE
            WRITE (6,*) 'BL: DETECTED MINIMUM ', ITER, ' ITERATIONS'
            WRITE (8,*) 'BL: DETECTED MINIMUM ', ITER, ' ITERATIONS'
         ENDIF
      ELSE
         WRITE (6,*) 'BL: TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'BL: TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE FITTWO(S,Y,A1,A2,D1,D2,S1,S2,BL,AVSUM,NLINES,NB)

C TO CALCULATE TWO-PEAK MODEL USING SUPPLIED PARAMETERS

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL PI
      REAL NS1, NS2, NI1, NI2, NI, NS, MFP, HEIGHT(1000), SUM
      INTEGER I, NLINES
      REAL DELTAY, AVSUM, YSUM
      REAL S1, S2, BL
      REAL A1, D1, C1
      REAL A2, D2, C2
      REAL NB

      NS1 = 0.0
      NS2 = 0.0
      NI1 = 0.0
      NI2 = 0.0
      NI = 0.0
      NS = 0.0
      MFP = 0.0
      SUM = 0.0
      DELTAY = 0.0
      AVSUM = 0.0
      YSUM = 0.0

      PI = 3.14
      C1 = D1*SQRT((2*PI))
      C2 = D2*SQRT((2*PI))

      DO 8 I = 1, NLINES

C NO OF SITES OF STRENGTH S

         IF ((S(I)-BL) .GT. 0) THEN
            NS1 = (A1/(C1*(S(I)-BL)))*EXP((-1.0/2.0)*
     &       (((LOG(S(I)-BL)-S1)/D1)**2))
         ELSE
            NS1 = 0.0
         ENDIF
         NS2 = (A2/C2)*EXP((-1.0/2.0)*(((S(I)-S2)/D2)**2))
         NS = NS1 + NS2

C NO OF SITES OF STRENGTH GREATER OR EQUAL TO S

         IF ((S(I)-BL) .GT. 0) THEN
            NI1 = (A1/2.0)*ERFC((LOG(S(I)-BL)-S1)/(D1*SQRT(2.0)))
         ELSE
            NI1 = 0
         ENDIF
         NI2 =(A2/2.0)*ERFC((S(I)-S2)/(D2*SQRT(2.0)))
         NI = NI1 + NI2

C MEAN FREE PATH
```

```
         IF (NI .NE. 0) THEN
            MFP = (1/NI)**(1.0/3.0)

C PEAK HEIGHT

            HEIGHT(I) = (NS*MFP) + NB
         ELSE
            HEIGHT(I) = NB
         ENDIF

C LEAST-SQUARES FIT

         DELTAY = HEIGHT(I) - Y(I)
         SUM = SUM + (DELTAY**2)
         YSUM = YSUM + Y(I)**2

         WRITE(1,*) S(I), Y(I), HEIGHT(I)
    8 CONTINUE
      REWIND(1)
      AVSUM = SQRT(SUM/YSUM)
      END

      SUBROUTINE ALIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)

C TO FIND BEST-FIT VALUE OF A FOR SINGLE PEAK - NORMAL DISTRIBUTION

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A, D, SV, AVSUM
      INTEGER NLINES, ITER, LIM1, LIM2
      REAL AU, AD, AVSUMU, AVSUMD, NB

      ITER = 0

    4 CALL FITLIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)
      AU = A + (A/100)
      CALL FITLIN(S, Y, AU, D, SV, AVSUMU, LIM1, LIM2, NLINES, NB)
      AD = A - (A/100)
      CALL FITLIN(S, Y, AD, D, SV, AVSUMD, LIM1, LIM2, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            A = AD
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            A = AU
            ITER = ITER + 1
            GOTO 4
         ENDIF
      ELSE
         WRITE (6,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE DLIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)

C TO FIND BEST-FIT VALUE OF D FOR SINGLE PEAK - NORMAL DISTRIBUTION

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A, D, SV, AVSUM
      INTEGER NLINES, ITER, LIM1, LIM2
      REAL DU, DD, AVSUMU, AVSUMD, NB

      ITER = 0
```

```
    4 CALL FITLIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)
      DU = D + (D/100)
      CALL FITLIN(S, Y, A, DU, SV, AVSUMU, LIM1, LIM2, NLINES, NB)
      DD = D - (D/100)
      CALL FITLIN(S, Y, A, DD, SV, AVSUMD, LIM1, LIM2, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            D = DD
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            D = DU
            ITER = ITER + 1
            GOTO 4
         ENDIF
      ELSE
         WRITE (6,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE FITLIN(S, Y, A, D, SV, AVSUM, LIM1, LIM2, NLINES, NB)

C TO FIT MODEL TO A SINGLE, NORMAL-DISTRIBUTION PEAK

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A, D, C, SV
      REAL NS, NI, MFP, HEIGHT(1000), DELTAY, SUM, PI
      INTEGER I, NLINES, LIM1, LIM2
      REAL AVSUM, NB, YSUM

      PI = 3.14
      C = D*SQRT((2*PI))

      NS = 0.0
      NI = 0.0
      MFP = 0.0
      DELTAY = 0.0
      SUM = 0.0
      AVSUM = 0.0
      YSUM = 0.0

    3 DO 2 I = LIM1, LIM2
         NS = (A/C)*EXP((-1.0/2.0)*(((S(I)-SV)/D)**2))
         NI =(A/2.0)*ERFC((S(I)-SV)/(D*SQRT(2.0)))
         MFP = (1/NI)**(1.0/3.0)
         HEIGHT(I) = (NS*MFP) + NB
         DELTAY = HEIGHT(I) - Y(I)
         SUM = SUM + (DELTAY**2)
         YSUM = YSUM + (Y(I)**2)
    2 CONTINUE
      REWIND(1)
      AVSUM = SQRT(SUM/YSUM)
      RETURN
      END

      SUBROUTINE ALOG(S, Y, A, D, SV, BL, AVSUM, LIM1, LIM2, NLINES, NB)

C TO FIND BEST-FIT VALUE OF A FOR SINGLE PEAK - LOG-NORMAL DISTRIBUTION

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A, D, SV, AVSUM
      INTEGER NLINES, ITER, LIM1, LIM2
      REAL AU, AD, AVSUMU, AVSUMD, NB, BL
```

```
      ITER = 0

    4 CALL FITLOG(S, Y, A, D, BL, SV, AVSUM, LIM1, LIM2, NLINES, NB)
      AU = A + (A/100)
      CALL FITLOG(S, Y, AU, D, BL, SV, AVSUMU, LIM1, LIM2, NLINES, NB)
      AD = A - (A/100)
      CALL FITLOG(S, Y, AD, D, BL, SV, AVSUMD, LIM1, LIM2, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            A = AD
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            A = AU
            ITER = ITER + 1
            GOTO 4
         ENDIF
      ELSE
         WRITE (6,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE DLOG(S, Y, A, D, SV, BL, AVSUM, LIM1, LIM2, NLINES, NB)

C TO FIND BEST-FIT VALUE OF D FOR SINGLE PEAK - LOG-NORMAL DISTRIBUTION

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A, D, SV, AVSUM
      INTEGER NLINES, ITER, LIM1, LIM2
      REAL DU, DD, AVSUMU, AVSUMD, NB, BL

      ITER = 0

    4 CALL FITLOG(S, Y, A, D, BL, SV, AVSUM, LIM1, LIM2, NLINES, NB)
      DU = D + (D/100)
      CALL FITLOG(S, Y, A, DU, BL, SV, AVSUMU, LIM1, LIM2, NLINES, NB)
      DD = D - (D/100)
      CALL FITLOG(S, Y, A, DD, BL, SV, AVSUMD, LIM1, LIM2, NLINES, NB)
      IF (ITER .LE. 10000) THEN
         IF ((AVSUMD .LT. AVSUM) .AND. (AVSUM .LT. AVSUMU)) THEN
            D = DD
            ITER = ITER + 1
            GOTO 4
         ELSE IF ((AVSUMD .GT. AVSUM) .AND. (AVSUM .GT. AVSUMU)) THEN
            D = DU
            ITER = ITER + 1
            GOTO 4
         ENDIF
      ELSE
         WRITE (6,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
         WRITE (8,*) 'TOO MANY ITERATIONS; PROCESS FAILED'
      ENDIF
      RETURN
      END

      SUBROUTINE FITLOG(S,Y,A,D,BL,SV,AVSUM,LIM1,LIM2,NLINES,NB)

C TO FIT MODEL TO A SINGLE, NORMAL-DISTRIBUTION PEAK

      IMPLICIT NONE
      REAL S(1000), Y(1000)
      REAL A, D, C, SV
      REAL NS, NI, MFP, HEIGHT(1000), DELTAY, SUM, PI, BL
      INTEGER I, NLINES, LIM1, LIM2
```

```
      REAL AVSUM, NB, YSUM

      PI = 3.14
      C = D*SQRT((2*PI))

      NS = 0.0
      NI = 0.0
      MFP = 0.0
      DELTAY = 0.0
      SUM = 0.0
      AVSUM = 0.0
      YSUM = 0.0

    3 DO 2 I = LIM1, LIM2
      IF ((S(I)-BL) .GT. 0) THEN
         NS = (A/(C*(S(I)-BL)))*EXP((-1.0/2.0)*
     &    (((LOG(S(I)-BL)-SV)/D)**2))
         NI =(A/2.0)*ERFC((LOG(S(I)-BL)-SV)/(D*SQRT(2.0)))
         MFP = (1/NI)**(1.0/3.0)
         HEIGHT(I) = (NS*MFP) + NB
         DELTAY = HEIGHT(I) - Y(I)
         SUM = SUM + (DELTAY**2)
         YSUM = YSUM + Y(I)**2
      ELSE
         HEIGHT(I) = NB
      ENDIF
    2 CONTINUE
      REWIND(1)
      AVSUM = SQRT(SUM/YSUM)
      RETURN
      END
```