A FORTRAN PROGRAM FOR ESTIMATING PARAMETERS IN A CUMULATIVE DIS(ETC)

S J BROWN, N HEUSER, P N SOMERVILLE

MAR 81

AFOSR-TR-81-0120

F19628-80-C-0004

UNCLASSIFIED

SCIENTIFIC-3
A FORTRAN PROGRAM FOR ESTIMATING PARAMETERS IN A CUMULATIVE DISTRIBUTION FUNCTION

Steven J. Bean
Mark Heuser
Paul N. Somerville

Scientific Report No. 3

31 March 1981

Approved for public release; distribution unlimited
Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.
The report documents, and gives a listing for a FORTRAN program written
to estimate the parameters of a cumulative distribution function which best
fits an empirical cumulative distribution function in a least squares sense.
Non-linear regression techniques are used. The program as listed uses the
Weibull distribution for the fit, but with the replacement of certain modules
the program may be used to fit many different distributions.
TABLE OF CONTENTS

1. Introduction 2
2. Flow Chart 3
3. FORTRAN Code 4
4. Some Remarks About the Program 15
5. Sample Output 15
6. References 15
1. **Introduction**

Given a large set of measurements of some quantity or variable, it is often useful to model the data using some statistical distribution function. For example if one has records of the "visibility" at Mildenhall, England for 10 a.m. February over a number of years, one may fit a Weibull distribution to the data. The Weibull distribution has two parameters, and the values selected for the two parameters are the ones for which the model best fits the data.

This report documents a FORTRAN program that has been written to estimate the parameters of a statistical distribution function which best fits a set of measurements on some variable. The fit is "best" in the sense that the model cumulative distribution function and the empirical cumulative distribution function (from the data) are closest to each other in the least squares sense.

Suppose the measurements are ordered from smallest to largest, that is $x(1) \leq x(2) \leq x(3) \leq \ldots \leq x(N)$ where $x(i)$ represents the $i$th smallest measurement. Then the empirical cumulative distribution function may be defined as

$$
\hat{F}_N(x) = \frac{2i-1}{2N} \quad \text{for} \quad x(1) \leq x < x(i+1)
$$

$$
= 0 \quad \text{for} \quad x < x(1) . \quad (1.1)
$$

If $F(x; \theta)$ is the model cumulative distribution function, then the values for $\theta$ ($\theta$ may be a vector of values) which are chosen are those which minimize the expression

$$
L = \sum_{i=1}^{N} \left[ \frac{(2i-1)/(2N)-F(x_i; \theta)}{2} \right]^2 . \quad (1.2)
$$

In the FORTRAN program, the determination of $\theta$, is accomplished by non-linear regression techniques where $F_N(x) = (2i-1)/(2N)$ for $i = 1, 2, \ldots, N$, is the dependent variable. Expression (1.2), the quantity to be minimized, is the "Residual Sum of Squares". A detailed description of the techniques used to fit distributions to data using non-linear regression techniques is given in Heuser, Somerville and Bean (1980).
2. **Flow Chart**

In non-linear regression, the model may be written

\[ y = F(x; \theta) + \epsilon, \quad (2.1) \]

where \( \theta \) represents a vector of unknown parameters. The usual technique is to linearize \( F(x; \theta) \) by the use of a first order Taylor Series expansion about guessed values \( \theta_0 \). The expression (2.1) is then linear in \( (\theta - \theta_0) \), and the usual least squares regression methods may be used to estimate \( \theta - \theta_0 \), the "correction" to the original guessed value. The procedure is then repeated with a first order Taylor Series expansion about the "corrected" guessed value for \( \theta \), the process terminating when the percentage reduction in residual sum of squares is less than some specified amounts. The flow chart below outlines the program.
3. FORTRAN Code

C*******************************************************************************
C
C TITLE: WEIBULL NONLINEAR REGRESSION PROGRAM
C THE FOLLOWING PROGRAM REGRESSES VISIBILITY DATA ON THE
C WEIBULL DISTRIBUTION.
C
C INPUT: N, X(1:N), ACC, AND Y(1:N) (SEE VARIABLE DICTIONARY)
N, X(1:N), AND ACC ARE INPUT ONCE IN THE BEGINNING OF THE
PROGRAM. Y(1:N) IS INPUT ONCE FOR EACH REGRESSION.
C
C OUTPUT: SUMMARY STATISTICS OF EACH REGRESSION INCLUDING ALPHA, BETA,
MONTH, HOUR, SID, RMS, COUNT, X(1:N), Y(1:N), PREDICTED
VALUES OF THE DISTRIBUTION, AND THE RESIDUALS.
C
C SUBROUTINES: GUESS, SSE, WEIBULL, FSSEA, PSSEE, SUCCESS, FAIL,
SECANT, CORVEC, MODLIN
C
C FOR A GENERAL OVERVIEW OF THE REGRESSION PROBLEM, SEE 'LEAST
SQUARES FITTING OF DISTRIBUTIONS USING NON-LINEAR REGRESSION' BY
MARK HEUSER, PAUL SOMERVILLE, AND STEVE BEAN.
C
C*******************************************************************************
C
C*******************************************************************************

  VARIABLE DICTIONARY

C
C N:  THE NUMBER OF OBSERVATIONS (MAXIMUM OF 15)
C X(1:N):  THE VALUES OF THE INDEPENDENT VARIABLE (DISTANCE)
C Y(1:N):  THE OBSERVED VISIBILITY PROBABILITIES AT EACH X
C ALPHA,BETA:  THE PARAMETERS IN THE WEIBULL MODEL
C STARTA,STARTB:  THE STARTING VALUES FOR ALPHA AND BETA COMPUTED BY
  THE SUBROUTINE ‘GUESS’
C CORA,CORB:  THE CORRECTION VECTORS FOR ALPHA AND BETA COMPUTED BY
  THE SUBROUTINE ‘CORVEC’
C NRSS,ORSS:  THE RSS FOR TWO CONSECUTIVE ESTIMATES OF ALPHA AND
  BETA. NRSS IS FROM THE NEWER ESTIMATE; ORSS IS FROM
  THE OLDER ESTIMATE.
C MONTH,HOUR,SID:  MONTH, HOUR, AND STATION IDENTIFIERS
C COUNT:  A LOOP COUNTER
C CONVERGE:  A LOGICAL VARIABLE THAT INDICATES CONVERGENCE.
C ACC:  AN INTEGER VALUE CONTROLLING THE ACCURACY OF THE STARTING
  VALUES FOR ALPHA AND BETA. SEE SUBROUTINE ‘GUESS’.  
C
C*******************************************************************************
REAL ALPHA,BETA,X(15),Y(15),STARTA,STARTB,NRSS,ORSS,CORA,CORB
REAL SSE,WEIBUL,RMS
INTEGER SID,MONTH,HOUR,N,COUNT,ACC
LOGICAL CONVERGE
COMMON N,X,Y
WRITE(6,200) ! PRINT A TITLE
READ,N ! INPUT THE NUMBER OF OBSERVATIONS
READ,(X(I),I=1,N) ! INPUT VALUES OF THE INDEPENDENT VARIABLE
READ,ACC ! INPUT LEVEL OF ACCURACY OF STARTING VALUES
C
! THE FOLLOWING LOOP INPUTS AND REGRESSIONS ON THE EMPIRICAL
! DISTRIBUTION. THE LOOP (AND THE PROGRAM) TERMINATES ON
! END OF FILE.
C
READ(5,100,END=40) (Y(I),I=1,N),SID,MONTH,HOUR
C
CALL GUESS(STARTA,STARTB,ACC) ! GET STARTING VALUES FOR ALPHA
ALPHA=STARTA ! AND BETA
BETA=STARTB
NRSS=SSE(ALPHA,BETA)
C
! THE FOLLOWING LOOP SOLVES FOR ALPHA AND BETA. CONVERGENCE IS
! ASSUMED WHEN THE PROPORTIONAL CHANGE IN THE RSS FOR TWO CON- 
! SECUTIVE ESTIMATES IS LESS THAN 1E-7.
C
COUNT=0 ! INITIALIZE THE LOOP CONTROL
CONVERGE=.FALSE. ! VARIABLES
C
20 IF ((COUNT.GT.50).OR.(CONVERGE)) GOTO 30
ORSS=NRSS
CALL CORVEC(ALPHA,BETA,CORA,CORB)
CALL MODLIN(ALPHA,BETA,CORA,CORB)
NRSS=SSE(ALPHA,BETA)
CONVERGE=ABS(ORSS-NRSS).LT.(ORSS*1.0E-7)
COUNT=COUNT+1
GOTO 20
C
30 IF (CONVERGE) THEN
   CALL SUCCESS(SID,MONTH,HOUR,ALPHA,BETA,NRSS,COUNT)
ELSE
   CALL FAIL(SID,MONTH,HOUR,STARTA,STARTB,ALPHA,BETA,NRSS)
END IF
C
GOTO 10
C
40 STOP
C
100 FORMAT(1X,14F4.3,15,I2,11)
200 FORMAT(/,35X,'NONLINEAR REGRESSION OF THE WEIBULL MODEL ON ',
& 'VISIBILITY DATA'/)
END
C

CWEIBUL IS A REAL FUNCTION THAT COMPUTES THE VALUE OF THE WEIBULL
C DISTRIBUTION FOR THE INPUT PARAMETERS X, ALPHA, AND BETA. ALL
C COMMUNICATION WITH THE PROCEDURE IS THROUGH THE PARAMETER LIST
C AND FUNCTION NAME.

C******************************************************************************

REAL FUNCTION WEIBUL(X,ALPHA,BETA)
   REAL X,ALPHA,BETA
   WEIBUL=1.0-EXP(-ALPHA*(X**BETA))
   RETURN
END

C******************************************************************************

C SSE IS A REAL FUNCTION THAT COMPUTES THE SUM OF THE SQUARED ERRORS
C IN THE WEIBULL MODEL AS A FUNCTION OF ALPHA AND BETA. COMMUNICATION
C WITH THE PROCEDURE IS DONE THROUGH THE PARAMETER LIST, THE
C FUNCTION NAME, AND THE COMMON BLOCK.

C******************************************************************************

REAL FUNCTION SSE(ALPHA,BETA)
   INTEGER I,N
   REAL ALPHA,BETA,X(15),Y(15),WEIBUL
   COMMON N,X,Y
   SSE=0.0
   DO 10 I=1,N
      SSE=SSE+(Y(I)-WEIBUL(X(I),ALPHA,BETA))**2
   10 CONTINUE
   RETURN
END

C******************************************************************************

C CORVEC IS A SUBROUTINE THAT COMPUTES THE CORRECTION VECTORS CORA AND
C CORE. COMMUNICATION WITH THE PROCEDURE IS DONE THROUGH THE
C PARAMETER LIST AND THE COMMON BLOCK. THE INPUT PARAMETERS ARE
C ALPHA AND BETA; OUTPUT PARAMETERS ARE CORA AND CORB.

C******************************************************************************

SUBROUTINE CORVEC(ALPHA,BETA,CORA,CORB)
   INTEGER I,N
   REAL ALPHA,BETA,CORA,CORB
   REAL DERADERB,TEMFPRSWIEIBUL,C11,C12,C22,D1,D2
   COMMON N,X,Y
   
   END
C11=0.0  C12=0.0  C22=0.0  D1=0.0  D2=0.0

C11, C12, C22, D1, and D2 represent a symmetric system of 2 equations in 2 unknowns. The 2 unknowns are CORA and CORB. Here C11, C12, C22, D1, and D2 are initialized to 0. In the DO loop that follows, their values are computed.

DO 10 I=1,N
    RS=Y(I)-WEIBUL(X(I),ALPHA,BETA)  ! RS=OBS-EXP
    TEMP=X(I)**BETA
    DERA=TEMP*EXP(-ALPHA*TEMP)  ! DERIVATIVE WITH RESPECT TO ALPHA
    DERB=DERA*LOG(X(I))*ALPHA  ! DERIVATIVE WITH RESPECT TO BETA
    C11=C11+(DERA**2)  ! COMPUTE C11
    C12=C12+(DERA*DERB)  ! COMPUTE C12
    C22=C22+(DERB**2)  ! COMPUTE C22
    D1=D1+(DERA*RS)  ! COMPUTE D1
    D2=D2+(DERB*RS)  ! COMPUTE D2
CONTINUE

10 CONTINUE

TEMP=(C11*C22)-(C12**2)  ! NOW THE SYSTEM IS SOLVED USING CRAMER'S RULE

CORA=((D1*C22)-(D2*C12))/TEMP
CORB=((C11*D2)-(D1*C12))/TEMP
RETURN
END

******************************************************************************

MODLIN is a subroutine that implements the modification of the linearization method proposed by H.O. Hartley in his paper "The modified Gauss-Newton method for the fitting of non-linear regression functions by least squares." All communication with the procedure is done through the parameter list: ALPHA, BETA, CORA, CORB. MODLIN optimizes the correction vectors computed by CORVEC and then adds them to ALPHA and BETA. When the procedure returns, ALPHA and BETA are the new parameter estimates. The values of CORA and CORB may have been changed in the procedure.

******************************************************************************

SUBROUTINE MODLIN(ALPHA, BETA, CORA, CORB)
    REAL ALPHA, BETA, CORA, CORB
    REAL DD01, DD02, V, SSE, DENOM
    REAL TEMP

    LET THETA=(ALPHA, BETA) BE THE CURRENT PARAMETER VALUES AND
    DELTA=(CORA, CORB) BE THE CORRECTION VECTOR. MODLIN ESTIMATES
    THE VALUE OF V=0 THAT MINIMIZES SSE(THETA+V*DELTA). SSE IS
    COMPUTED AT THETA (00), THETA+.5*DELTA (01), AND THETA+DELTA
    (02). V IS FOUND SO THAT THETA+V*DELTA IS THE VERTEX OF THE
    PARABOLA PASSING THROUGH 00, 01, AND 02.
Q0=SSE(ALPHA,BETA)
Q1=SSE(ALPHA+.5*CORA,BETA+.5*CORB)
Q2=SSE(ALPHA+CORA,BETA+CORB)

DENOM=4.0*(Q2+Q0-(2.0*Q1))

! IF DENOM IS CLOSE TO ZERO THEN WE CAN'T COMPUTE V WITHOUT
! PRODUCING A DIVIDE-BY-ZERO OR FLOATING-POINT-OVERFLOW
! ERROR. IN THIS CASE, ADD THE CORRECTION VECTOR ASSOCIATED
! WITH THE MINIMUM OF Q1 AND Q2 TO ALPHA AND BETA.

IF (ABS(DENOM).LT.1E-15) THEN
    IF (Q1.LT.Q2) THEN
        ALPHA=ALPHA+.5*CORA
        BETA=BETA+.5*CORB
    ELSE
        ALPHA=ALPHA+CORA
        BETA=BETA+CORB
    END IF
    RETURN
END IF

V=.5+((Q0-Q2)/DENOM)
TEMP=SSE(ALPHA+V*CORA,BETA+V*CORB)

! IF V<0 OR TEMP>Q0 THEN THE COMPUTATION OF V IS REDONE WITH
! DELTA=.5*DELTA.

IF (((V.LT.0.0).OR.(TEMP.GT.Q0)) THEN
    CORA=.5*CORA
    CORB=.5*CORB
    Q2=Q1
    Q1=SSE(ALPHA+.5*CORA,BETA+.5*CORB)
    GOTO 10
END IF

ALPHA=ALPHA+V*CORA
BETA=BETA+V*CORB
RETURN
END
SUBROUTINE GUESS(ALPHA,BETA,ACC)
  EXTERNAL PSSEA,PSSEB
  INTEGER I,J,N,E(3),ACC
  REAL ALPHA,BETA,C(3),T1,T2,PSSEA,PSSEB
  REAL X(15),Y(15)
  LOGICAL CONVERGE
  COMMON N,X,Y

  GIVEN TWO DATA POINTS IN THE EMPIRICAL DISTRIBUTION, WE CAN
  SOLVE FOR ALPHA AND BETA SO THAT THE WEIBULL MODEL FITS
  THROUGH THOSE TWO POINTS EXACTLY. GUESS CHOOSES THREE DATA
  POINTS AND FITS THE WEIBULL THROUGH THE FIRST TWO, THE LAST
  TWO, AND THE FIRST AND LAST POINTS, THUS ARRIVING AT THREE
  DIFFERENT ESTIMATES FOR ALPHA AND BETA. THE AVERAGES OF THE
  THREE ESTIMATES ARE USED AS STARTING VALUES FOR ALPHA AND
  BETA

  ALPHA=0.0
  BETA=0.0
  IF (Y(N),EQ.0.0) RETURN

  E(1)=2
  E(2)=8
  E(3)=13

  DO 20 I=1,3
     T(I)=X(E(I))
     IF (Y(E(I)),EQ.0.0) THEN
        C(I)=.00001
     ELSE
        C(I)=Y(E(I))
     END IF
     CONTINUE

  DO 30 I=1,3
     T=MOD(I,3)+1
     T1=ALOG(ALOG(1.0-C(I))/ALOG(1.0-C(J)))/ALOG(D(I)/D(J))
     BETA=BETA+T1
     ALPHA=ALPHA+(-ALOG(1.0-C(I))/D(I)**T1)
     CONTINUE

  ALPHA=ALPHA/3.0
  BETA=BETA/3.0

  MOST OF THE TIME THESE STARTING VALUES WILL BE GOOD ENOUGH TO
  BEGIN THE NONLINEAR REGRESSION PROCEDURE. SOME CASES
  HOWEVER, WILL REQUIRE EVEN MORE ACCURATE STARTING VALUES.
  THE FOLLOWING CODE OPTIONALLY IMPROVES THE STARTING VALUES
  DEPENDING ON THE VALUE OF ACC. ACC IS MERELY THE
  NUMBER OF TIMES THE LOOP BELOW IS EXECUTED. THE LOOP TRIES
  TO OPTIMIZE ALPHA FOR A FIXED BETA, AND THEN OPTIMIZES BETA
  FOR A FIXED ALPHA.
IF (ACC.LE.0) RETURN

DO 40 I=1,ACC
    T1=ALPHA
    T2=BETA
    CALL SECANT(ALPHA,BETA,ALPHA,PSSEB,CONVERGE)
    IF (CONVERGE) CALL SECANT(ALPHA,BETA,BETA,PSSEB,CONVERGE)
    IF (.NOT.(CONVERGE)) GOTO 50
40 CONTINUE
RETURN

ENDr

C***********************************************************************
C F$AIL IS AN OUTPUT ROUTINE CALLED WHEN A DISTRIBUTION FAIL$ TO CON-
C VERGE AFTER 50 ITERATIONS. THE VALUES OF SEVERAL VARIABLES ARE
C WRITTEN TO THE OUTPUT FILE. COMMUNICATION WITH THE PROCEDURE IS
C DONE THROUGH THE PARAMETER LIST AND THE COMMON BLOCK. ALL PARA-
C METERS ARE INPUT PARAMETERS.
C***********************************************************************

SUBROUTINE FAIL(SID,MNTH,HOUR,STARTA,STARTB,ALPHA,BETA,NRSP)
INTEGER SID,MNTH,HOUR,N,1
REAL STARTA,STARTB,ALPHA,BETA,NRSP,TMP,X(15),Y(15),SSE
COMMON N,Y,Y

S$E(SSE(STARTA,STARTB))
WRITE(6,100)
WRITE(6,200) SID,MNTH,HOUR
WRITE(6,300)
WRITE(6,400) STARTA,STARTB,TMP
WRITE(6,500) ALPHA,BETA,NRSP
WRITE(6,600)
WRITE(6,700)
DO 10 I=1,N
    WRITE(6,800) X(I),Y(I)
10 CONTINUE
RETURN

100 FORMAT(1X,T32,35(* '*'))
200 FORMAT(1X,'ATTENTION: STATION ','I2',' MONTH ','I2',' HOUR '-'
   I1,' FAILED TO ',/I1,' CONVERGE AFTER 50 ITERATIONS.'))
300 FORMAT(1X,'A VARIABLE DUMP FOLLOWS:'))
400 FORMAT(1X,'STARTA='G15.7,1X,'STARTB='G15.7,1X,'NRSP='G15.7,1X)
500 FORMAT(1X,'ALPHA='G15.7,1X,'BETA='G15.7,1X)
600 FORMAT(1X,'SSE(ALPHA,BETA)'='G15.7,1X,'SSE(STARTA,STARTB)'='G15.7)
700 FORMAT(1X,'ENDPTS',I4X,'OBSCUMFR')
800 FORMAT(1X,'****',14X,'**')
900 FORMAT(1X,G15.7,9X,G15.7)
END
SUCCESS IS AN OUTPUT ROUTINE CALLED WHEN A DISTRIBUTION CONVERGES SUCCESSFULLY WITHIN 50 ITERATIONS. SUMMARY STATISTICS OF THE REGRESSION ARE WRITTEN TO THE OUTPUT FILE. COMMUNICATION WITH THE PROCEDURE OCCURS THROUGH THE PARAMETER LIST AND COMMON BLOCK. ALL PARAMETERS ARE INPUT PARAMETERS.

SUBROUTINE SUCCESS(SID, MONTH, HOUR, ALPHA, BETA, NRSS, COUNT)
    INTEGER SID, MONTH, HOUR, N, COUNT
    REAL ALPHA, BETA, NRSS, RMS, X(15), Y(15), EX, RS, WEIBUL
    COMMON H, X, Y

    RMS=SQR(NRSS/FLOAT(N))
    WRITE(6,100)
    WRITE(6,200)
    WRITE(6,300)
    WRITE(6,400) SID, MONTH, HOUR, ALPHA, BETA, RMS, COUNT
    WRITE(5,500)
    WRITE(6,600)
    DO 10 I=1,N
        EX=WEIBUL(X(I), ALPHA, BETA)
        RS=Y(I)-EX
        WRITE(6,700) X(I), Y(I), EX, RS
    CONTINUE
    RETURN

FORMAT(100,T31,35(* ' '),/)
200 FORMAT(10X,'STATION ID',5X,'MONTH',5X,'HOUR',10X,'ALPHA',-15X, 
      'BETA',17X,'RMS',11X,'OF ITERATIONS')
300 FORMAT(09X,' ',5X,' ',10X,' ',15X, 
      ' ',17X,' ',11X, /)
400 FORMAT(12X,16,2(8X,12-E),6XG15.7,4XG14.7,4XG14.7,11Y,13,/')
500 FORMAT(T38,'ENDPTS',5X,'OCUMFR',10X,'EXCUMFR',12,'RESIDUAL')
600 FORMAT(10X,' ',T38,' ',5X,' ',10X,' ',12Y, 
      ',/)
700 FORMAT(T37,F7.4,6X,F5.3,7X,G15.7,5X,G15.7)
END

SECANT IS A SUBROUTINE THAT USES THE SECANT METHOD OF ROOT SOLVING TO FIND THE ROOT OF PSSEA HOLDING BETA CONSTANT, OR TO FIND THE BEST ALPHA FOR A GIVEN BETA, OR THE BEST BETA FOR A GIVEN ALPHA. COMMUNICATION WITH THE PROCEDURE OCCURS THROUGH THE PARAMETER LIST AND THE COMMON BLOCK. IN THE PARAMETER LIST, ALPHA AND BETA ARE THE CURRENT VALUES OF THE MODEL PARAMETERS. FARM IS THE VARIABLE BEING OPTIMIZED (EITHER ALPHA OR BETA), AND PDER IS THE PARTIAL DERIVATIVE FUNCTION (EITHER PSSEA OR PSSEB). CONVERGE IS A LOGICAL VARIABLE INDICATING WHETHER THE SECANT METHOD CONVERGED ON A ROOT. TO SOLVE FOR THE ROOT OF PSSEA, SET FARM=ALPHA AND PDER=PSSEA. THE OPTIMIZED VALUE OF ALPHA WILL BE RETURNED. TO SOLVE FOR THE ROOT OF PSSEB, SET FARM=BETA AND PDER=PSSEB. THE OPTIMIZED VALUE OF BETA WILL BE RETURNED.
SUBROUTINE SECANT(ALPHA, BETA, PARM, PDER, CONVERGE)
REAL ALPHA, BETA, PARM, PDER, X(15), Y(15), T1, T2, T3, DELTA
INTEGER N, I
LOGICAL CONVERGE
COMMON N, X, Y

T2 = PDER(ALPHA, BETA)
DELTA = 0.001
PARM = PARM + DELTA
CONVERGE = .TRUE.

DO 10 I = 1, 15
   T1 = PDER(ALPHA, BETA)
   T3 = T1 - T2
   IF (ABS(T3), LE, 1E-15) GOTO 15
   DELTA = (-T1*DELTA)/(T3)
   PARM = PARM + DELTA
   IF (ABS(Delta), LE, 1E-7) GOTO 20
   T2 = T1
10  CONTINUE
15  CONVERGE = .FALSE.
20  RETURN
END

************************************************************************************
FSSEA is a real function that computes the partial derivative of SSE
with respect to ALPHA. Communication with the procedure occurs
through the function name, common block, and parameter list. ALPHA
and BETA are input parameters; the value of the derivative is
returned through the function name.

real function FSSEA(ALPHA, BETA)
real ALPHA, BETA, X(15), Y(15), T1, T2
INTEGER N, I
COMMON N, X, Y

FSSEA = 0.0
DO 10 I = 1, N
   T1 = X(I)**BETA
   T2 = EXP(-ALPHA*T1)
   FSSEA = FSSEA + (Y(I) - 1.0 + T2)*(-T1*T2)
10  CONTINUE

RETURN
END
SUBROUTINE PSSEB

REAL FUNCTION PSSEB(ALPHA,BETA)

REAL ALPHA,BETA,X(15),Y(15),T1,T2,T3,T4
INTEGER N,I
COMMON N,X,Y

PSSEB=0.0
DO 10 I=1,N
   T1=X(I)**BETA
   T2=EXP(-ALPHA*T1)
   T3=ALPHA*LOG(X(I))*T1*T2
   T4=Y(I)-1.0+T2
   PSSEB=PSSEB+T3*T4
10 CONTINUE
RETURN
END
NONLINEAR REGRESSION OF THE WEIBULL MODEL ON VISIBILITY DATA

<table>
<thead>
<tr>
<th>STATION ID</th>
<th>MONTH</th>
<th>HOUR</th>
<th>ALPHA</th>
<th>BETA</th>
<th>RMS</th>
<th># OF ITERATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.3001646</td>
<td>0.6776246</td>
<td>0.2175113E-01</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMDIS</th>
<th>ODCUMER</th>
<th>EXCMER</th>
<th>RESIDUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2500</td>
<td>0.110</td>
<td>0.1107035</td>
<td>-0.7034689E-03</td>
</tr>
<tr>
<td>0.3125</td>
<td>0.159</td>
<td>0.1275731</td>
<td>0.3142692E-01</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.202</td>
<td>0.1711044</td>
<td>0.3089564E-01</td>
</tr>
<tr>
<td>0.6250</td>
<td>0.216</td>
<td>0.1961113</td>
<td>0.1988868E-01</td>
</tr>
<tr>
<td>0.7500</td>
<td>0.229</td>
<td>0.2188597</td>
<td>0.1014027E-01</td>
</tr>
<tr>
<td>1.0000</td>
<td>0.235</td>
<td>0.2593037</td>
<td>-0.2430369E-01</td>
</tr>
<tr>
<td>1.2500</td>
<td>0.276</td>
<td>0.2947218</td>
<td>-0.1872179E-01</td>
</tr>
<tr>
<td>1.5000</td>
<td>0.305</td>
<td>0.3263727</td>
<td>-0.2137274E-01</td>
</tr>
<tr>
<td>2.0000</td>
<td>0.355</td>
<td>0.3812875</td>
<td>-0.2828747E-01</td>
</tr>
<tr>
<td>2.5000</td>
<td>0.434</td>
<td>0.4279263</td>
<td>0.6973684E-02</td>
</tr>
<tr>
<td>3.0000</td>
<td>0.437</td>
<td>0.4684348</td>
<td>-0.3143480E-01</td>
</tr>
<tr>
<td>4.0000</td>
<td>0.553</td>
<td>0.5360345</td>
<td>0.1696545E-01</td>
</tr>
<tr>
<td>5.0000</td>
<td>0.618</td>
<td>0.5906984</td>
<td>0.2730155E-01</td>
</tr>
<tr>
<td>6.0000</td>
<td>0.642</td>
<td>0.6360627</td>
<td>0.5937338E-02</td>
</tr>
</tbody>
</table>
4. Some Remarks About the Program

The program given in this report was written to fit the Weibull distribution to visibility data. The visibility data was that contained in the "Revised Uniform Summary of Weather Observations" (RUSSWO's) prepared by the Data Processing Division of the Air Weather Service. Some changes must be made to fit another distribution to another variable. The program is made up of a series of subroutines and functions so that these may be altered to fit the users need without changing the flow of the program.

The function WEIBUL must be replaced by the desired cumulative distribution function (CDF). Also, the name of the function, WEIBUL, must be changed throughout the program if the name of the function is changed. The subroutine GUESS which gives the initial values of the parameters must be changed to correspond to the distribution being used. The same basic idea can be used. However, it may take more programming for other distributions particularly if the distribution is not in closed form. The subroutine CORVEC must be changed where the partial derivatives DERA and DERB occur. Another change is required in the functions PSSEA and PSSEB which are functions which take partials with respect to each of the parameters of the sum of squared errors. The appropriate partials must replace those of the Weibull distribution in each of these functions.

The program was designed to run in a batch environment, and the rules of standard FORTRAN - IV were adhered to as closely as possible. The only departures from FORTRAN - IV are the use of the IF-THEN-ELSE-ENDIF structure commonly found in FORTRAN 77, and the use of the exclamation mark to permit comments on the same line as code. Both deviations were made merely for clarity's sake in the program.

5. Sample Output

A sample output is shown in page 14.

6. Reference
