NBSIR 80-2151

A New Method of Assigning Uncertainty in Volume Calibration

James A. Lechner, Charles P. Reeve, Clifford H. Spiegelman

Issued December 1980

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NBSIR 80-2151

A NEW METHOD OF ASSIGNING UNCERTAINTY IN VOLUME CALIBRATION

James A.Lechner, Charles P. Reeve, Clifford H. Spiegelman

Issued December 1980

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, *Secretary*

Jordan J. Baruch, *Assistant Secretary for Productivity, Technology, and Innovation*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

*A NEW METHOD OF ASSIGNING UNCERTAINTY IN VOLUME CALIBRATION

by

James A. Lechner
Charles P. Reeve
Clifford H. Spiegelman

with programming assistance from
Martin Ross Cordes and Janice M. Knapp

*Work supported (in part) by the U. S. Nuclear Regulatory Commission.

ABSTRACT

This paper presents a practical statistical overview of the pressure-volume calibration curve for large nuclear materials processing tanks. It explains the appropriateness of applying splines (piecewise polynomials) to this curve, and it presents an overview of the associated statistical uncertainties. In order to implement these procedures a practical and portable FORTRAN IV program is provided along with its users' manual. Finally, the recommended procedure is demonstrated on actual tank data collected by NBS.

Key Words: Volume calibration; differential pressure; splines; accountability; statistics.

TABLE OF CONTENTS

1. Introduction.

The direct measurement of liquid volume in large processing tanks, especially with internal structure, is impractical at best. Measuring (differential) pressure is simple and quick. However, in order to estimate the volume indirectly by observing pressure, it is necessary to use the relationship between volume and pressure. This relationship is known as a <u>calibration curve</u>; its <u>estimation</u> is the process known as calibration.

Fitting a calibration curve is much like regression, in that for "known" values v_i of volume, one obtains one or more observations p_{ij} of the differential pressure $p_i = p(v_i)$, and "fits" a response function $p(v)$ by statistical methods - usually by least squares. At this point, the correspondence stops. Whereas regression is used to predict values of the dependent variable (p) for given values of the independent variable (v), or to test a proposed relationship between the variables, a calibration curve is used to estimate values of the independent variable v corresponding to new measured values of the dependent variable p. Furthermore, the confidence interval (or uncertainty measure) which is desired is not for p, but rather for v. And finally, systematic error is introduced by lack of fit of the calibration curve, and in a materials accounting situation this may be crucial.

This paper presents a method for producing valid uncertainty limits for the pressure-volume tank calibration curve by using calibration functions which are smooth, piecewise polynomial functions called "splines." Taking advantage of an approach to calibration originated by Scheffé [1] and further elucidated by Scheffé, Rosenblatt and

Spiegelman [2], it provides statistically sound uncertainty limits, not just for a single estimated value of volume, but for all volumes estimated by use of the fitted curve. This approach overcomes a major theoretical problem with earlier methods: it makes proper allowance for the contribution to the overall uncertainty of errors in fitting the curve.

The procedure presented herein has been implemented, based upon a spline-fitting program due to deBoor [3]. The resulting FORTRAN program has been tested on various sets of data, including actual tank data.

The remainder of this paper is organized as follows. Section 2 contains a discussion of the pressure-volume model, and the statistics of calibration. Section 3 contains a discussion of an example, and of the printout produced by the program. Section 4 is essentially a users' manual for the program. In Section 5 will be found a discussion of open questions, work in progress, cautions, and possible extensions of this technique. Finally, Appendix 1 contains a discussion of Scheffe's constant c, used as an input to the program, and Appendix 2 contains a listing of the program.

2. The Scientific Basis for, and Interpretation of, a Calibration Curve.

In order to provide statistically valid uncertainty limits for the volume estimates obtained through the use of a calibration curve, it is necessary to have a prior model for the pressure-volume relationship. That is, while the constants in the model for the relationship may be determined from the data, the form of the model must not depend on the data to be used in fitting the relationship. In addition, the less accurately the hypothesized model describes this relationship, the less valid will be the resulting uncertainty statements. That is, inaccuracies in the hypothesized model will lead to systematic differences between true and fitted curves. In order to obtain valid uncertainty statements, bounds for such differences must be determined, and added to the statistical uncertainty as systematic error limits.

The interiors of large processing tanks do not generally conform to idealized geometrical shapes, such as cylinders. Often, however, the tank can be considered to be composed only of segments for which an idealized model is a good representation. In this paper it is assumed that the tank is composed of a finite number, k+1, of distinct and known regions where the idealized relationship between the two variables pressure, p, and volume, v, is given by

$$
\begin{aligned}
p &= f(v) \\
&= g_1(v) \qquad \xi_0 < v < \xi_1 \\
&= g_2(v) \qquad \xi_1 < v < \xi_2 \\
&\;\vdots \\
&= g_{k+1}(v) \qquad \xi_k < v < \xi_{k+1} .
\end{aligned}
$$

In addition, continuity of the relationship at the interior "knots" ξ_i, $i=1,\ldots,k$ is required.

In all that follows, volume and height refer to the portion of the tank above the bottom of the diptube used to measure pressure. The portion of the tank below that point is known as the heel, and is not treated in this paper. The pressure measured is the difference in pressure between the bottom of the diptube and a reference point at the top of the tank.

The pressure-volume relationship can be ascertained from the following two equations. At height h the volume in the container is

$$v = \int_0^h A(x)\, dx$$

where A(x) is the cross-sectional area at height x. Also, when the liquid height is h, $p = h\rho g$, where ρ is the density of the homogeneous liquid and g is the acceleration due to gravity. Using these two equations one easily obtains

$$v = \int_0^{p/\rho g} A(x)\, dx \, .$$

Thus $\frac{\partial v}{\partial p} = \frac{1}{\rho g} A(p/\rho g)$ and hence in areas of the tank where A(x) is constant the volume-pressure relationship is a straight line.

If A(x) is constant* in region i, as it obviously is for at least some regions of the tank shown in Figure 1, then

(1) $$p = g_i(v) = \gamma_i + \beta_i v \text{ for } \xi_{i-1} < v < \xi_i \, , \; i=1,\ldots,k+1.$$

We assume that each pressure measurement has a random error associated with it, and that these errors are independent and

* If A(x) is not constant on an interval, then p is not a linear function of v on that interval. The program under discussion uses the B-spline basis when higher-order polynomial splines are required, because as pointed out in reference [4], the use of simpler representations of polynomial splines may lead to numerical instability.

normally-distributed, with mean zero and constant variance σ^2. (Recall that volume is assumed to be measured with no significant error.) Because of these errors, only estimates of the coefficients γ_i and β_i are obtained during the calibration process (experiment). These coefficient estimates are then used during plant operation to obtain estimates of the volume in the tank, utilizing the inverse of the relationship (1).

Determination of uncertainty limits on these estimates is not trivial. There are two sources of random error: estimation of the coefficients γ and β in the calibration experiment, and measurement of p during operational use of the tank. The familiar linear regression model has properties, such as the nonexistence of means and variances of reciprocals, which make the analysis difficult. Special justification involving asymptotic (large sample-size) behavior is thus required in order to use a propagation-of-error approach to obtain appropriate approximate uncertainty limits on the estimated volumes. Furthermore, unless the p-v relationship is linear, normally distributed errors in the p-measurements during operation produce non-normal errors in the resulting estimates of v. The usual propagation-of-error technique does not take into account the differing characteristics of these errors. The new technique presented in this paper, in contrast to the propagation-of-error approach just mentioned, does allow a correct accounting for both.

The calibration chart (i.e., the table of uncertainty limits) is produced after choosing two probabilities, α and δ. An exact statement giving the interpretation of these probabilities may be found in Scheffé

[1] and in Scheffé, Rosenblatt, and Spiegelman [2]. However, an expanded, more heuristic explanation is given here. First, we require bounds for the calibration curve which will contain the entire curve with a prechosen probability $1-\delta$. (Thus, δ can be thought of as describing the uncertainty level to be associated with the outcome of the <u>initial calibration experiment</u>.) These bounds guarantee, with probability $1-\delta$, that for any and every future volume v within the range of calibration of the tank, the v-interval (see Figure 2) that would be obtained by projection of the value f(v) through the curves to $\underline{v}$ and $\overline{v}$ would contain v. The second probability level to be chosen is α. (Here α can be thought of as describing the uncertainty level to be attributed to errors in <u>future</u> individual pressure measurements.) If σ were known, we could state that the true pressure f(v) at the unknown volume v lies within the $1-\alpha$ confidence interval $(p-z_{1-\alpha/2}\sigma,\ p + z_{1-\alpha/2}\sigma)$ with probability $1-\alpha$, where p is the observed pressure and $z_{1-\alpha/2}$ is the two-sided $1-\alpha$ value for a normal distribution. The Scheffé procedure expands this interval appropriately, to account for the facts that σ is estimated and that this estimate is used for the $1-\delta$ bound on the curve and for bounds on many different f(v). It then combines the $1-\alpha$ confidence interval for f(v) with the $1-\delta$ bounds on the calibration curve to produce calibration intervals I(p) for v. Construction of the calibration intervals is shown schematically in Figure 3. A set of intervals $I(p_i)$ for p_i in the range of values obtained during the calibration experiment is called a calibration chart (see Figure 4).

In the discussion of the example presented in the next section, more detail on the nature of the steps that make up a calibration run will be found.

3. An Example.

This example relates to a processing tank, roughly circular in cross section, but with internal structure consisting of cooling coils, stirrers, braces, etc. [5]. This tank is pictured in Figure 1. The data from calibration runs on this tank have graciously been made available by the author of reference [5].

There were five calibration runs for which the data were useful for this analysis. One run was done in the canyon where the tank is to be used. The other four, done in a mock-up area, used smaller tubing in the pressure-measuring system. This smaller tubing was known to cause systematic differences in the pressure measurement, which were expected to be linearly related to pressure for each run. Since the tubing in the canyon was sufficiently large to render the pressure drop insignificant, the systematic error was estimated for each of the other four runs, and a correction made by applying a linear transformation to the measured pressure. It should be noted that these corrections, made to four of the five runs, effectively decrease the degrees of freedom for the error sum of squares by eight (two correction parameters times four runs corrected).

The calibration program was applied to these data, as were various other techniques available on the large central computer at NBS. The results will now be presented and their use described.

In the version of the program described here, the knot locations are input by the analyst. It is presumed that the knot locations can be adequately prescribed from the blueprints and other knowledge about the tank. (A refinement which allows the automatic determination of the number and location of knots is being investigated.) The program displays the given knot locations and other input data, as shown in Figure 5. Next come the results of the fitting operation, as shown in part in Figure 6. Note that the fit here is a fit of observed pressure (y) as a function of the accurately-dispensed volume (x); it is pressure which is subject to errors of observation, and volume which is essentially known. The residual standard deviation, an estimate of the standard deviation of the pressure measurements, is derived from the residuals or deviations of the measured pressures from the fitted curve. In this case, its value is 1.49 pascals. This value, the corresponding degrees of freedom, and the coefficients (which are in general not immediately interpretable, since they refer to the so-called B-splines, a representation chosen for computational stability), are part of these results. The calibration intervals for estimation of volume from measured pressure are printed next (see Figure 4). An ordinary polynomial representation and a residual plot are also printed as shown in Figures 7 and 8.

It will be instructive to examine the printout and discuss the approach in more detail, and this will now be done.

As can be seen from Figure 5, the program duplicates the end knots. This is just a simple way to define the B-spline basis functions which are used to perform the fit, and need not concern the analyst. The input values for knot locations, degree of fit, and other miscellaneous parameters are printed out for verification.

At this point, the program does a linear least squares fit of the specified model to the (v,p) data, and prints out a reasonably standard summary (Figure 6). The column labels are self-explanatory. At the bottom of this summary are found the residual standard deviation and its associated degrees of freedom, and the estimated coefficients with the corresponding estimated standard deviations.

The program next computes some intermediate results which generally are of no interest to the analyst, and therefore are only printed out if requested. These are confidence intervals for p, at 300 evenly-spaced points covering the range of v between the extreme knots. Input values of α, δ, and c are used in this procedure, so these values are printed.

The calibration chart comes next, giving the predicted value of v and the corresponding lower and upper limits for each of the specified set of p-values (see Figure 4). It is obtained by inverse interpolation from the confidence intervals for p discussed in the preceding paragraph. Usually, the extreme values of p will be at least partially outside the range of at least one of the curves. When this happens, the intervals should extend either to zero volume or to full volume. This is indicated by "<" and ">" respectively on the printout.

Since the coefficients actually fitted are the B-spline coefficients, the program converts the B-spline representation to a simple polynomial representation. The printout shows the endpoints and the coefficients of the fitted polynomial for each of the specified intervals (see Fig. 7).

Finally, the residuals from the fitted model are plotted in order of increasing volume to allow a visual check of the adequacy of the chosen model [6] (see Fig. 8).

At NBS, with the aid of the central computer and the OMNITAB system [7], a number of other things were tried which strengthen the conviction that this program does indeed work well. These will now be discussed.

Various subsets of the data were fitted to the same model. No inconsistencies were found.

The sensitivity to position and presence of the different knots was checked. The results were rather sensitive to the knot locations, which implies that good estimates of the locations are required for good fits. It should also be noted that where a knot bounds a short segment, the removal of that knot might make very little difference in any global measure of fit quality, unless there are many data points in that short stretch. Nevertheless, the systematic error introduced by deleting that knot can be a consistent source of inventory losses or gains, apparent or real. Thus it is important to include all real segments in the model to be fitted.

A separate program was written to perform linear spline fitting, while the main package was being put together. The answers did not differ between the two programs, providing a partial check that no programming errors were committed.

Smooth higher-order spline fits were tried (quadratic and cubic). There was no improvement in fit. The linear spline appears to provide an adequate representation of the pressure/volume relationship.

Probability plots were done in various ways, looking for possible troubles with the data or the method. Nothing suspicious was found.

4. User's Manual.

This fixed-knot spline package for calibration consists of a "main" subroutine SPLEEN and 29 additional subroutines. The manner in which they interact is diagrammed in Figure 9. All programs are written in FORTRAN and have been checked for portability by the Bell Laboratories PFORT verifier [8]. It was decided that SPLEEN should be a subroutine rather than a main program so that the user could enter the parameter values in the way most convenient for him. The user then must write a main program which sets up the required dimensioned variables and assigns values to the necessary parameters (those with asterisks in the list which follows). These parameters are passed to subroutine SPLEEN via the statement

```
CALL SPLEEN(H,X,Y,W,R1,R2,RES,N,NX,NKX,T,BCOEF,XXI,Q,DIAG,K,
            KX,YY,NY,NYX,MD,SCRTCH,JX,AL,DL,C,IP)
```

where

* H(80) = Up to 80 characters in 80A1 format identifying the data

* X(NX) = Vector (length N) of X-values where observations were made (independent variable)

* Y(NX) = Vector (length N) of observations

* W(NX) = Vector (length N) of weights for observations

R1(NKX) = Vector (length N+K) for scratch area

R2(NKX) = Vector (length N+K) for scratch area

RES(NKX) = Vector (length N+K) of residuals from spline fit

* N = Number of observations

* NX = Dimension (≥N) of vectors X,Y,W

* NKX = Dimension (≥N+K) of vectors R1,R2,RES

* T(KX) = Vector (length K+2*MD) of knot locations

BCOEF(KX) = Vector (length K+MD-1) of B-spline coefficients

XXI(KX,KX) = Variance-covariance matrix (size [K+MD-1]×[K+MD-1]) of B-spline coefficients

Q(JX,KX) = Matrix (size [MD+1]×[K+MD-1]) for scratch area

DIAG(KX) = Vector (length K+MD-1) for scratch area

* K = Number of knots specified by user (later increased to K+2*MD by program)

* KX = Dimension (>K+2*MD) of vectors T,BCOEF,DIAG and matrices XXI and Q (column only for Q)

* YY(NYX) = Vector (length NY) of Y-values for which predicted X-values (with confidence intervals) are to be computed

* NY = Number of Y-values for which predicted X-values are to be computed

* NYX = Dimension (>NY) of vector YY

* MD = Degree of spline (<19); for example, 1=linear, 2=quadratic, 3=cubic)

SCRTCH(JX,JX) = Matrix (size [MD+1]×[MD+1]) for scratch area

* JX = Dimension of square matrix SCRTCH and row dimension of matrix Q = 20

* AL = Alpha level of significance

* DL = Delta level of significance

* C = Constant in the interval (0.85,1.25) associated with Scheffé's calibration technique (see Appendix 1 for a discussion of this constant)

* IP =
- 0 For abbreviated printout
- 1 For full printout (residuals, PP representation)
- 2 For full printout plus Y-confidence intervals for 300 evenly spaced X-values over knot span

Variables which appear with an asterisk (*) require input values from the main program. The subscripts on vectors and matrices indicate the dimensions which must be assigned in the main program.

Variable names which begin with the letters I,J,K,L,M, or N are of the INTEGER type. The remaining variable names are of the REAL single precision type.

The print parameter IP gives the user a certain amount of control over the amount of information to be printed out. Normally the most suitable value is IP=1. A value of IP=0 suppresses the printout of the weights, independent variable, observations, predicted values, and residuals. This option may save quite a bit of paper in case there are several hundred observations, but it deprives the user of the chance to visually examine the residuals. A value of IP=2 causes a listing of certain intermediate vectors which are somewhat lengthy and would not normally be of use to the user.

In the interest of minimizing the number of variables needed in the CALL statement, not all of the printed information can be recovered through the passed parameters. Furthermore, three of the variables (X, K, and T) return values different from their input values.

The data points (X_i, Y_i, W_i) may be input in arbitrary order, as may the knot locations T_i and the vector of YY_i specifying the y-values on the calibration chart.

There are two subroutines which check for consistency among the input parameters. Each inconsistency causes a diagnostic message to be printed. If one or more inconsistencies is detected then the program execution is terminated. Observations outside the knot span are flagged and weighted zero. The number of observations is then reduced by one for each flagged point and a diagnostic is printed. This is not a fatal error unless it reduces the number of degrees of freedom to zero or less.

Although this package can handle splines of any degree up to 19 it was primarily intended for splines of lower degree, i.e., linear, quadratic, or cubic. Test runs on sets of both real and artificial data have given valid results up to about degree 9. Beyond that the limitation of single precision arithmetic on the 36-bit NBS central computer begins to cause roundoff errors that invalidate the results. The user should exercise caution when fitting the higher degree splines.

If the user wants to change some of the continuity conditions at a given knot he may do so by duplicating that knot in the knot vector which is passed to subroutine SPLEEN. If a knot appears M times in the fitting of a spline of degree N then the functional value and the first N-M derivatives of the function will be continuous at that knot. If M = N+1, neither the function nor its derivatives are required to be continuous.

The package may be applied to both monotone increasing and decreasing calibration curves.

5. Summary and Discussion.

An approach to calibration curves and their uncertainty bands has been presented, complete with a FORTRAN program to perform the required calculations. An example involving a large process tank has been used to illustrate the approach and the program. The results include not only the curve for estimating volume from measured pressure, but also valid uncertainty limits for repeated applications of the calibration curve obtained.

The interval estimates of volume comprise two parts: a long-term component which changes only at recalibration, and a random component.

The contribution due to the long-term component may be estimated in large scale calibration experiments by the volume interval estimate obtained when $\alpha=1$. Similarly, the contribution due to the random component may be estimated by the volume interval estimate obtained when $\delta=1$. When the calibration experiment is of a more modest size involving less than 100 data points the above component estimates may not be realistic. However a more comprehensive treatment for combining interval estimates (and hence their components) obtained from a calibration curve is under development by C. Spiegelman and K. Eberhardt [9].

The results of a calibration will be used repeatedly, usually without any further opportunity to verify their correctness, until the next calibration. Therefore it is important that the measurement system be under control. In the work reported here, the run-to-run differences observed in the mock-up area were due to a known source (the small diameter of the tubing), and could be corrected. If any anomalous behavior is observed which cannot be satisfactorily explained, then of course the entire statistical analysis must be approached with caution.

Little has been written about the design of calibration experiments - i.e., the selection of volumes at which pressure is to be measured, the number of measurements to be taken at each volume, and the arrangement of these measurement points into a sequence of runs. One solution to this question has been achieved by Spiegelman and Studden, and will be published in the NBS Journal of Research [10]. In general, later runs will concentrate on certain sections of a tank, but it is good practice to ensure that at least two runs cover each section, and that several runs cover the whole tank. If this precaution is not taken, there might be very little cross-validation between runs.

Certain caveats ought to be mentioned here. The program under discussion assumes that the knot locations are known, and that the model is correct. Consequences of failure of these assumptions could be severe. With respect to the knot locations, careful inspection of the residual plots will sometimes indicate discrepancies. These may be small; however, it is important to realize that such regions represent systematic deviations, and could be used (at least in theory) to cover the diversion of material. An approach to the problem involving unknown knot locations is being pursued at this writing.

Unlike a simple linear regression, where the inclusion of a superfluous higher-order term generally causes no major trouble (the fitted coefficient turns out insignificantly small, and the residual mean square increases minimally), choosing a higher-order model when fitting smooth splines can result in a very much worse fit. This is because of the smoothing restrictions, which greatly limit the freedom of the fitting procedure. (Imagine that the true model consists of two straight lines meeting at a point. If one chooses to fit a quadratic spline, then one is insisting on having two quadratic curves which meet at the proper x-value, <u>and</u> which have the same slope at that point. Thus the slope at that point is probably going to be some value between the two straight-line slopes, and the fit cannot be accurate.) One way around this difficulty is to fit piecewise polynomials (i.e., do not require smoothness), and investigate the appropriate degree from these fits. However, it is much better to know the situation well enough to choose the correct model from physical considerations.

A technique that the authors have found useful is to run the program described here with the degree of the fit set equal to zero. The result is to fit a step function to the data, and to produce a plot of the residuals from that fit; for the Example of this paper, that plot is reproduced as Figure 10. It can be seen that the residuals look as linear as a printer plot can look. Therefore, a first-degree (linear) spline fit is the proper choice. If in some segment the relationship were not linear, this plot should show it. The plot also gives some idea of the spread of points across the intervals, though of course near-duplicate points will plot as one because it is a discrete printer plot.

Note that the continuity restrictions can be relaxed when using the program under consideration, by simply duplicating the knots. See Section 4 for details.

Appendix 1.

As stated in Section 4, a constant c must be input by the user. In order to obtain this constant from tables 1 and 2 in Scheffé (1973) for $1 < p < 10$ the user must have calculated the standard deviation, SD, for $\hat{p}(v)$ in the complete region of calibration. The smallest and largest values the $SD(\hat{p}(v))/\hat{\sigma}$ over the complete calibration region are used as input for the Scheffé tables. For $k > 10$ Scheffé gives a mathematical algorithm for finding c, and states that for very large (asymptotic) values of n-k, c=1. (Here n is the number of observations, and k is the number of B-spline coefficients.) If the reader does not wish to do a Scheffé table 1 or table 2 lookup, the following table gives approximate and generally larger values for this constant.

Approximate c values
for $1 < k < 10$

n-k	60-119	120-149	150 +
c	1.10	1.05	1.00

Appendix 2.

Program Listing.

The subroutines which make up the spline-fitting package follow, in alphabetical order.

CPR*NS(1).ADKNTS(2)

```
      SUBROUTINE ADKNTS (T,K,KX,MO)                                     ADKNTS01
C-----------------------------------------------------------------------ADKNTS02
C     ADKNTS   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING     ADKNTS03
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. ADKNTS04
C              AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION ADKNTS05
C     FOR: DUPLICATING THE FIRST AND KTH (LAST) ENTRIES OF THE KNOT     ADKNTS06
C          VECTOR T (MO-1) TIMES                                        ADKNTS07
C     SUBPROGRAMS CALLED: -NONE-                                        ADKNTS08
C     CURRENT VERSION COMPLETED OCTOBER 10, 1979                        ADKNTS09
C-----------------------------------------------------------------------ADKNTS10
      DIMENSION T(KX)                                                   ADKNTS11
   10 FORMAT (//1X,29(1H-)/1X,29H* SUMMARY OF KNOT LOCATIONS */1X,      ADKNTS12
     2 29(1H-)//5X,1HI,6X,8HKNOTS(I)/)                                  ADKNTS13
   20 FORMAT (2X,I4,G15.6)                                              ADKNTS14
   30 FORMAT (/5X,30H<<<<< EACH END KNOT DUPLICATED,I3,1X,              ADKNTS15
     2 11HTIMES >>>>>)                                                  ADKNTS16
C--- SAVE END KNOT LOCATIONS                                            ADKNTS17
      Q1=T(1)                                                           ADKNTS18
      Q2=T(K)                                                           ADKNTS19
C--- INCREASE INDEX OF EACH KNOT LOCATION BY (MO-1)                     ADKNTS20
      KM=K+MO                                                           ADKNTS21
      DO 40 I=1,K                                                       ADKNTS22
      KMI=KM-I                                                          ADKNTS23
      KI=K-I+1                                                          ADKNTS24
      T(KMI)=T(KI)                                                      ADKNTS25
   40 CONTINUE                                                          ADKNTS26
C--- ADD DUPLICATE END KNOT LOCATIONS AT THEIR RESPECTIVE ENDS          ADKNTS27
      MD=MO-1                                                           ADKNTS28
      DO 50 I=1,MD                                                      ADKNTS29
      KMI=KM+I-1                                                        ADKNTS30
      T(I)=Q1                                                           ADKNTS31
      T(KMI)=Q2                                                         ADKNTS32
   50 CONTINUE                                                          ADKNTS33
C--- RECOMPUTE THE LENGTH OF THE VECTOR T                               ADKNTS34
      K=K+2*MD                                                          ADKNTS35
      WRITE (6,30) MD                                                   ADKNTS36
C--- WRITE NEW VECTOR OF KNOT LOCATIONS                                 ADKNTS37
      WRITE (6,10)                                                      ADKNTS38
      DO 60 I=1,K                                                       ADKNTS39
      WRITE (6,20) I,T(I)                                               ADKNTS40
   60 CONTINUE                                                          ADKNTS41
      RETURN                                                            ADKNTS42
      END                                                               ADKNTS43
```

CPR*NS(1).BCHFAC(2)

```
      SUBROUTINE BCHFAC (W,NBNDMX,NBANDS,NROW,DIAG)                     BCHFAC01
C  FROM  * A PRACTICAL GUIDE TO SPLINES *  BY C. DE BOOR                BCHFAC02
CONSTRUCTS CHOLESKY FACTORIZATION                                       BCHFAC03
C                     C  =  L * D * L-TRANSPOSE                         BCHFAC04
C  WITH L UNIT LOWER TRIANGULAR AND D DIAGONAL, FOR GIVEN MATRIX C OF   BCHFAC05
C  ORDER  N R O W , IN CASE  C  IS (SYMMETRIC) POSITIVE SEMIDEFINITE    BCHFAC06
C  AND  B A N D E D , HAVING  N B A N D S  DIAGONALS AT AND BELOW THE   BCHFAC07
C  MAIN DIAGONAL.                                                       BCHFAC08
C                                                                       BCHFAC09
C******  I N P U T  ******                                              BCHFAC10
C  NROW.....IS THE ORDER OF THE MATRIX  C .                             BCHFAC11
C  NBNDMX.....THE ACTUAL ROW DIMENSION OF W.                            BCHFAC12
C  NBANDS.....INDICATES ITS BANDWIDTH, I.E.,                            BCHFAC13
C          C(I,J) = 0 FOR ABS(I-J) .GT. NBANDS .                        BCHFAC14
C  W.....WORKARRAY OF SIZE (NBANDS,NROW)  CONTAINING THE  NBANDS  DIAGO-BCHFAC15
C        NALS IN ITS ROWS, WITH THE MAIN DIAGONAL IN ROW  1 . PRECISELY,BCHFAC16
C        W(I,J)  CONTAINS  C(I+J-1,J), I=1,...,NBANDS, J=1,...,NROW.    BCHFAC17
C          FOR EXAMPLE, THE INTERESTING ENTRIES OF A SEVEN DIAGONAL SYM-BCHFAC18
C        METRIC MATRIX  C  OF ORDER  9  WOULD BE STORED IN  W  AS       BCHFAC19
C                                                                       BCHFAC20
C                        11 22 33 44 55 66 77 88 99                     BCHFAC21
C                        21 32 43 54 65 76 87 98                        BCHFAC22
C                        31 42 53 64 75 86 97                           BCHFAC23
C                        41 52 63 74 85 96                              BCHFAC24
C                                                                       BCHFAC25
C        ALL OTHER ENTRIES OF  W  NOT IDENTIFIED IN THIS WAY WITH AN EN-BCHFAC26
C        TRY OF  C  ARE NEVER REFERENCED .                              BCHFAC27
C  DIAG.....IS A WORK ARRAY OF LENGTH  NROW .                           BCHFAC28
C                                                                       BCHFAC29
C******  O U T P U T  ******                                            BCHFAC30
C  W.....CONTAINS THE CHOLESKY FACTORIZATION  C = L*D*L-TRANSP, WITH    BCHFAC31
C        W(1,I) CONTAINING  1/D(I,I)                                    BCHFAC32
C        AND  W(I,J)  CONTAINING  L(I-1+J,J), I=2,...,NBANDS.           BCHFAC33
C                                                                       BCHFAC34
C******  M E T H O D  ******                                            BCHFAC35
C   GAUSS ELIMINATION, ADAPTED TO THE SYMMETRY AND BANDEDNESS OF  C , ISBCHFAC36
C   USED .                                                              BCHFAC37
C     NEAR ZERO PIVOTS ARE HANDLED IN A SPECIAL WAY. THE DIAGONAL ELE-  BCHFAC38
C  MENT C(N,N) = W(1,N) IS SAVED INITIALLY IN  DIAG(N), ALL N. AT THE N-BCHFAC39
C  TH ELIMINATION STEP, THE CURRENT PIVOT ELEMENT, VIZ.  W(1,N), IS COM-BCHFAC40
C  PARED WITH ITS ORIGINAL VALUE, DIAG(N). IF, AS THE RESULT OF PRIOR   BCHFAC41
C  ELIMINATION STEPS, THIS ELEMENT HAS BEEN REDUCED BY ABOUT A WORD     BCHFAC42
C  LENGTH, (I.E., IF W(1,N)+DIAG(N) .LE. DIAG(N)), THEN THE PIVOT IS DE-BCHFAC43
C  CLARED TO BE ZERO, AND THE ENTIRE N-TH ROW IS DECLARED TO BE LINEARLYBCHFAC44
C  DEPENDENT ON THE PRECEDING ROWS. THIS HAS THE EFFECT OF PRODUCING    BCHFAC45
C   X(N) = 0  WHEN SOLVING  C*X = B  FOR  X, REGARDLESS OF  B. JUSTIFIC-BCHFAC46
C  ATION FOR THIS IS AS FOLLOWS. IN CONTEMPLATED APPLICATIONS OF THIS   BCHFAC47
C  PROGRAM, THE GIVEN EQUATIONS ARE THE NORMAL EQUATIONS FOR SOME LEAST-BCHFAC48
C  SQUARES APPROXIMATION PROBLEM, DIAG(N) = C(N,N) GIVES THE NORM-SQUAREBCHFAC49
C  OF THE N-TH BASIS FUNCTION, AND, AT THIS POINT,  W(1,N)  CONTAINS THEBCHFAC50
C  NORM-SQUARE OF THE ERROR IN THE LEAST-SQUARES APPROXIMATION TO THE N-BCHFAC51
C  TH BASIS FUNCTION BY LINEAR COMBINATIONS OF THE FIRST N-1 . HAVING   BCHFAC52
C  W(1,N)+DIAG(N) .LE. DIAG(N) SIGNIFIES THAT THE N-TH FUNCTION IS LIN- BCHFAC53
C  EARLY DEPENDENT TO MACHINE ACCURACY ON THE FIRST N-1 FUNCTIONS, THEREBCHFAC54
C  FORE CAN SAFELY BE LEFT OUT FROM THE BASIS OF APPROXIMATING FUNCTIONSBCHFAC55
C     THE SOLUTION OF A LINEAR SYSTEM                                   BCHFAC56
C                        C*X = B                                        BCHFAC57
```

```
C   IS EFFECTED BY THE SUCCESSION OF THE FOLLOWING  T W O  CALLS:          BCHFAC58
C     CALL BCHFAC (W, NBNDMX, NBANDS, NROW, DIAG) , TO GET FACTORIZATIONBCHFAC59
C     CALL BCHSLV (W, NBNDMX, NBANDS, NROW, DIAG) , TO SOLVE FOR X.       BCHFAC60
C  THE VECTOR B NOW CONTAINS X.                                           BCHFAC61
C                                                                         BCHFAC62
C  MODIFICATION BY.                                                       BCHFAC63
C                                                                         BCHFAC64
C     MARTIN CORDES                                                       BCHFAC65
C     CENTER FOR APPLIED MATHEMATICS, NBS                                 BCHFAC66
C     VERSION 1                                                           BCHFAC67
C     OCT 1979                                                            BCHFAC68
C                                                                         BCHFAC69
C-------------------------------------------------------------------------BCHFAC70
      INTEGER NBNDMX,NBANDS,NROW,I,IMAX,J,JMAX,N                          BCHFAC71
      REAL W(NBNDMX,NROW),DIAG(NROW),RATIO                                BCHFAC72
      IF (NROW.GT.1) GO TO 10                                             BCHFAC73
      IF (W(1,1).GT.0.) W(1,1)=1./W(1,1)                                  BCHFAC74
      RETURN                                                              BCHFAC75
C                                         STORE DIAGONAL OF  C  IN  DIAG.BCHFAC76
   10 DO 20 N=1,NROW                                                      BCHFAC77
   20 DIAG(N)=W(1,N)                                                      BCHFAC78
C                                                         FACTORIZATION .BCHFAC79
      DO 70 N=1,NROW                                                      BCHFAC80
      IF (W(1,N)+DIAG(N).GT.DIAG(N)) GO TO 40                             BCHFAC81
      DO 30 J=1,NBANDS                                                    BCHFAC82
   30 W(J,N)=0.                                                           BCHFAC83
      GO TO 70                                                            BCHFAC84
   40 W(1,N)=1./W(1,N)                                                    BCHFAC85
      IMAX=MIN0(NBANDS-1,NROW-N)                                          BCHFAC86
      IF (IMAX.LT.1) GO TO 70                                             BCHFAC87
      JMAX=IMAX                                                           BCHFAC88
      DO 60 I=1,IMAX                                                      BCHFAC89
      RATIO=W(I+1,N)*W(1,N)                                               BCHFAC90
      DO 50 J=1,JMAX                                                      BCHFAC91
      L1=N+I                                                              BCHFAC92
      L2=J+I                                                              BCHFAC93
   50 W(J,L1)=W(J,L1)-W(L2,N)*RATIO                                       BCHFAC94
      JMAX=JMAX-1                                                         BCHFAC95
   60 W(I+1,N)=RATIO                                                      BCHFAC96
   70 CONTINUE                                                            BCHFAC97
      RETURN                                                              BCHFAC98
      END                                                                 BCHFAC99
```

```
CPR*NS(1).BCHSLV(1)
      SUBROUTINE BCHSLV (W,NBNDMX,NBANDS,NROW,B)                        BCHSLV01
C  FROM  * A PRACTICAL GUIDE TO SPLINES *  BY C. DE BOOR                BCHSLV02
C  SOLVES THE LINEAR SYSTEM     C*X = B   OF ORDER  N R O W  FOR  X     BCHSLV03
C  PROVIDED  W  CONTAINS THE CHOLESKY FACTORIZATION FOR THE BANDED (SYM-BCHSLV04
C  METRIC) POSITIVE DEFINITE MATRIX  C  AS CONSTRUCTED IN THE SUBROUTINEBCHSLV05
C    B C H F A C  (QUO VIDE).                                           BCHSLV06
C                                                                       BCHSLV07
C******  I N P U T  ******                                              BCHSLV08
C  NROW.....IS THE ORDER OF THE MATRIX  C .                             BCHSLV09
C  NBNDMX.....THE ACTUAL ROW DIMENSION OF W.                            BCHSLV10
C  NBANDS.....INDICATES THE BANDWIDTH OF  C .                           BCHSLV11
C  W.....CONTAINS THE CHOLESKY FACTORIZATION FOR  C , AS OUTPUT FROM    BCHSLV12
C          SUBROUTINE BCHFAC  (QUO VIDE).                               BCHSLV13
C  B.....THE VECTOR OF LENGTH  N R O W  CONTAINING THE RIGHT SIDE.      BCHSLV14
C                                                                       BCHSLV15
C******  O U T P U T  ******                                            BCHSLV16
C  B.....THE VECTOR OF LENGTH  N R O W  CONTAINING THE SOLUTION.        BCHSLV17
C                                                                       BCHSLV18
C******  M E T H O D  ******                                            BCHSLV19
C  WITH THE FACTORIZATION  C = L*D*L-TRANSPOSE  AVAILABLE, WHERE  L  IS BCHSLV20
C  UNIT LOWER TRIANGULAR AND  D  IS DIAGONAL, THE TRIANGULAR SYSTEM     BCHSLV21
C  L*Y = B  IS SOLVED FOR  Y (FORWARD SUBSTITUTION), Y IS STORED IN  B, BCHSLV22
C  THE VECTOR  D**(-1)*Y IS COMPUTED AND STORED IN  B, THEN THE TRIANG- BCHSLV23
C  ULAR SYSTEM  L-TRANSPOSE*X = D**(-1)*Y IS SOLVED FOR  X (BACKSUBSTIT-BCHSLV24
C  UTION).                                                              BCHSLV25
C                                                                       BCHSLV26
C  MODIFICATION BY.                                                     BCHSLV27
C                                                                       BCHSLV28
C      MARTIN CORDES                                                    BCHSLV29
C      CENTER FOR APPLIED MATHEMATICS, NBS                              BCHSLV30
C      VERSION 1                                                        BCHSLV31
C      OCT 1979                                                         BCHSLV32
C                                                                       BCHSLV33
C-----------------------------------------------------------------------BCHSLV34
C                                                                       BCHSLV35
      INTEGER NBNDMX,NBANDS,NROW,J,JMAX,N,NBNDM1                        BCHSLV36
      REAL W(NBNDMX,NROW),B(NROW)                                       BCHSLV37
      IF (NROW.GT.1) GO TO 10                                           BCHSLV38
      B(1)=B(1)*W(1,1)                                                  BCHSLV39
      RETURN                                                            BCHSLV40
C                                                                       BCHSLV41
C                 FORWARD SUBSTITUTION. SOLVE L*Y = B FOR Y, STORE IN B.BCHSLV42
   10 NBNDM1=NBANDS-1                                                   BCHSLV43
      DO 30 N=1,NROW                                                    BCHSLV44
      JMAX=MIN0(NBNDM1,NROW-N)                                          BCHSLV45
      IF (JMAX.LT.1) GO TO 30                                           BCHSLV46
      DO 20 J=1,JMAX                                                    BCHSLV47
      L=J+N                                                             BCHSLV48
   20 B(L)=B(L)-W(J+1,N)*B(N)                                           BCHSLV49
   30 CONTINUE                                                          BCHSLV50
C                                                                       BCHSLV51
C     BACKSUBSTITUTION. SOLVE L-TRANSP.X = D**(-1)*Y  FOR X, STORE IN B.BCHSLV52
      N=NROW                                                            BCHSLV53
   40 B(N)=B(N)*W(1,N)                                                  BCHSLV54
      JMAX=MIN0(NBNDM1,NROW-N)                                          BCHSLV55
      IF (JMAX.LT.1) GO TO 60                                           BCHSLV56
      DO 50 J=1,JMAX                                                    BCHSLV57
```

```
      L=J+N                                                             BCHSLV58
   50 B(N)=B(N)-W(J+1,N)*B(L)                                           BCHSLV59
   60 N=N-1                                                             BCHSLV60
      IF (N.GT.0) GO TO 40                                              BCHSLV61
      RETURN                                                            BCHSLV62
      END                                                               BCHSLV63
```

CPR*NS(1).BSPLPP(1)

```
      SUBROUTINE BSPLPP (T,BCOEF,N,K,SCRTCH,BREAK,COEF,L,KMX)          BSPLP001
C  FROM  * A PRACTICAL GUIDE TO SPLINES *  BY C. DE BOOR               BSPLP002
CALLS  BSPLVB                                                          BSPLP003
C                                                                      BSPLP004
CONVERTS THE B-REPRESENTATION  T, BCOEF, N, K  OF SOME SPLINE INTO ITS BSPLP005
C  PP-REPRESENTATION  BREAK, COEF, L, K .                              BSPLP006
C                                                                      BSPLP007
C******  I N P U T  ******                                             BSPLP008
C  T.....KNOT SEQUENCE, OF LENGTH  N+K                                 BSPLP009
C  BCOEF.....B-SPLINE COEFFICIENT SEQUENCE, OF LENGTH  N               BSPLP010
C  N.....LENGTH OF  BCOEF  AND  DIMENSION OF SPLINE SPACE  SPLINE(K,T) BSPLP011
C  K.....ORDER OF THE SPLINE                                           BSPLP012
C  KMX.....ROW DIMENSION OF ARRAYS COEF AND SCRTCH                     BSPLP013
C                                                                      BSPLP014
C  W A R N I N G  . . .  THE RESTRICTION   K .LE. KMAX (= 20)   IS IMPO-BSPLP015
C        SED BY THE ARBITRARY DIMENSION STATEMENT FOR  BIATX  BELOW, BUTBSPLP016
C        IS  N O W H E R E   C H E C K E D   FOR.                      BSPLP017
C                                                                      BSPLP018
C******  W O R K   A R E A  ******                                     BSPLP019
C  SCRTCH......OF SIZE  (KMX,K) , NEEDED TO CONTAIN BCOEFFS OF A PIECE BSPLP020
C        OF THE SPLINE AND ITS  K-1  DERIVATIVES                       BSPLP021
C                                                                      BSPLP022
C******  O U T P U T  ******                                           BSPLP023
C  BREAK.....BREAKPOINT SEQUENCE, OF LENGTH  L+1, CONTAINS (IN INCREAS- BSPLP024
C        ING ORDER) THE DISTINCT POINTS IN THE SEQUENCE  T(K),...,T(N+1)BSPLP025
C  COEF.....ARRAY OF SIZE (KMX,N), WITH  COEF(I,J) = (I-1)ST DERIVATIVE BSPLP026
C        OF SPLINE AT BREAK(J) FROM THE RIGHT                          BSPLP027
C  L.....NUMBER OF POLYNOMIAL PIECES WHICH MAKE UP THE SPLINE IN THE IN-BSPLP028
C        TERVAL  (T(K), T(N+1))                                        BSPLP029
C                                                                      BSPLP030
C******  M E T H O D  ******                                           BSPLP031
C     FOR EACH BREAKPOINT INTERVAL, THE  K  RELEVANT B-COEFFS OF THE    BSPLP032
C  SPLINE ARE FOUND AND THEN DIFFERENCED REPEATEDLY TO GET THE B-COEFFS BSPLP033
C  OF ALL THE DERIVATIVES OF THE SPLINE ON THAT INTERVAL. THE SPLINE ANDBSPLP034
C  ITS FIRST  K-1  DERIVATIVES ARE THEN EVALUATED AT THE LEFT END POINT BSPLP035
C  OF THAT INTERVAL, USING  BSPLVB  REPEATEDLY TO OBTAIN THE VALUES OF  BSPLP036
C  ALL B-SPLINES OF THE APPROPRIATE ORDER AT THAT POINT.               BSPLP037
C                                                                      BSPLP038
C     PARAMETER KMAX = 20                                              BSPLP039
C                                                                      BSPLP040
C  MODIFICATION BY.                                                    BSPLP041
C                                                                      BSPLP042
C     MARTIN CORDES                                                    BSPLP043
C     CENTER FOR APPLIED MATHEMATICS, NBS                              BSPLP044
C     VERSION 1                                                        BSPLP045
C     MAR 1980                                                         BSPLP046
C                                                                      BSPLP047
C----------------------------------------------------------------------BSPLP048
C                                                                      BSPLP049
      INTEGER K,L,N,I,J,JP1,KMJ,LEFT,LSOFAR                            BSPLP050
      REAL BCOEF(N),BREAK(1),COEF(KMX,1),T(1),SCRTCH(KMX,K),BIATX(20), BSPLP051
     2 DIFF,FKMJ,SUM                                                   BSPLP052
C    *                                      ,BIATX(KMAX),DIFF,FKMJ,SUMBSPLP053
C     DIMENSION BREAK(L+1),COEF(K,L),T(N+K)                            BSPLP054
CURRENT FORTRAN STANDARD MAKES IT IMPOSSIBLE TO SPECIFY THE LENGTH OF  BSPLP055
C  BREAK , COEF  AND  T  PRECISELY WITHOUT THE INTRODUCTION OF OTHERWISEBSPLP056
C  SUPERFLUOUS ADDITIONAL ARGUMENTS.                                   BSPLP057
```

```
      LSOFAR=0                                                          BSPLP058
      BREAK(1)=T(K)                                                     BSPLP059
      DO 60 LEFT=K,N                                                    BSPLP060
C                                  FIND THE NEXT NONTRIVIAL KNOT INTERVAL.BSPLP061
      IF (T(LEFT+1).EQ.T(LEFT)) GO TO 60                                BSPLP062
      LSOFAR=LSOFAR+1                                                   BSPLP063
      BREAK(LSOFAR+1)=T(LEFT+1)                                         BSPLP064
      IF (K.GT.1) GO TO 10                                              BSPLP065
      COEF(1,LSOFAR)=BCOEF(LEFT)                                        BSPLP066
      GO TO 60                                                          BSPLP067
C        STORE THE K B-SPLINE COEFF.S RELEVANT TO CURRENT KNOT INTERVAL BSPLP068
C                             IN  SCRTCH(.,1) .                         BSPLP069
   10 DO 20 I=1,K                                                       BSPLP070
      M=LEFT-K+I                                                        BSPLP071
   20 SCRTCH(I,1)=BCOEF(M)                                              BSPLP072
C                                                                       BSPLP073
C        FOR J=1,...,K-1, COMPUTE THE  K-J  B-SPLINE COEFF.S RELEVANT TOBSPLP074
C        CURRENT KNOT INTERVAL FOR THE J-TH DERIVATIVE BY DIFFERENCING  BSPLP075
C        THOSE FOR THE (J-1)ST DERIVATIVE, AND STORE IN SCRTCH(.,J+1) . BSPLP076
      DO 30 JP1=2,K                                                     BSPLP077
      J=JP1-1                                                           BSPLP078
      KMJ=K-J                                                           BSPLP079
      FKMJ=FLOAT(KMJ)                                                   BSPLP080
      DO 30 I=1,KMJ                                                     BSPLP081
      M1=LEFT+I                                                         BSPLP082
      M2=M1-KMJ                                                         BSPLP083
      DIFF=T(M1)-T(M2)                                                  BSPLP084
      IF (DIFF.GT.0.) SCRTCH(I,JP1)=((SCRTCH(I+1,J)-SCRTCH(I,J))/DIFF)*FBSPLP085
     2KMJ                                                               BSPLP086
   30 CONTINUE                                                          BSPLP087
C                                                                       BSPLP088
C        FOR  J = 0, ..., K-1, FIND THE VALUES AT  T(LEFT)  OF THE  J+1 BSPLP089
C        B-SPLINES OF ORDER  J+1  WHOSE SUPPORT CONTAINS THE CURRENT    BSPLP090
C        KNOT INTERVAL FROM THOSE OF ORDER  J  (IN  BIATX ), THEN COMB- BSPLP091
C        INE WITH THE B-SPLINE COEFF.S (IN SCRTCH(.,K-J) ) FOUND EARLIERBSPLP092
C        TO COMPUTE THE (K-J-1)ST DERIVATIVE AT  T(LEFT)  OF THE GIVEN  BSPLP093
C        SPLINE.                                                        BSPLP094
C           NOTE. IF THE REPEATED CALLS TO  BSPLVB  ARE THOUGHT TO GENE-BSPLP095
C        RATE TOO MUCH OVERHEAD, THEN REPLACE THE FIRST CALL BY         BSPLP096
C           BIATX(1) = 1.                                               BSPLP097
C        AND THE SUBSEQUENT CALL BY THE STATEMENT                       BSPLP098
C           J = JP1 - 1                                                 BSPLP099
C        FOLLOWED BY A DIRECT COPY OF THE LINES                         BSPLP100
C           DELTAR(J) = T(LEFT+J) - X                                   BSPLP101
C                  ......                                               BSPLP102
C           BIATX(J+1) = SAVED                                          BSPLP103
C        FROM  BSPLVB . DELTAL(KMAX)  AND  DELTAR(KMAX)  WOULD HAVE TO  BSPLP104
C        APPEAR IN A DIMENSION STATEMENT, OF COURSE.                    BSPLP105
C                                                                       BSPLP106
      CALL BSPLVB (T,1,1,T(LEFT),LEFT,BIATX)                            BSPLP107
      COEF(K,LSOFAR)=SCRTCH(1,K)                                        BSPLP108
      DO 50 JP1=2,K                                                     BSPLP109
      CALL BSPLVB (T,JP1,2,T(LEFT),LEFT,BIATX)                          BSPLP110
      KMJ=K+1-JP1                                                       BSPLP111
      SUM=0.                                                            BSPLP112
      DO 40 I=1,JP1                                                     BSPLP113
   40 SUM=BIATX(I)*SCRTCH(I,KMJ)+SUM                                    BSPLP114
   50 COEF(KMJ,LSOFAR)=SUM                                              BSPLP115
```

```
   60 CONTINUE                                                   BSPLP116
      L=LSOFAR                                                   BSPLP117
      RETURN                                                     BSPLP118
      END                                                        BSPLP119
```

```
CPR*NS(1).BSPLVB(1)
      SUBROUTINE BSPLVB (T,JHIGH,INDEX,X,LEFT,BIATX)                    BSPLVB01
C  FROM  * A PRACTICAL GUIDE TO SPLINES *  BY C. DE BOOR                BSPLVB02
CALCULATES THE VALUE OF ALL POSSIBLY NONZERO B-SPLINES AT  X  OF ORDER  BSPLVB03
C                                                                       BSPLVB04
C               JOUT  =  MAX( JHIGH , (J+1)*(INDEX-1) )                 BSPLVB05
C                                                                       BSPLVB06
C  WITH KNOT SEQUENCE  T .                                              BSPLVB07
C                                                                       BSPLVB08
C******  I N P U T  ******                                              BSPLVB09
C  T.....KNOT SEQUENCE, OF LENGTH  LEFT + JOUT  , ASSUMED TO BE NONDE-  BSPLVB10
C        CREASING.  A S S U M P T I O N . . . .                         BSPLVB11
C                       T(LEFT)  .LT.  T(LEFT + 1)   .                  BSPLVB12
C   D I V I S I O N  B Y  Z E R O  WILL RESULT IF  T(LEFT) = T(LEFT+1)  BSPLVB13
C  JHIGH,                                                               BSPLVB14
C  INDEX.....INTEGERS WHICH DETERMINE THE ORDER  JOUT = MAX(JHIGH,      BSPLVB15
C        (J+1)*(INDEX-1))  OF THE B-SPLINES WHOSE VALUES AT  X  ARE TO  BSPLVB16
C        BE RETURNED.  INDEX  IS USED TO AVOID RECALCULATIONS WHEN SEVE-BSPLVB17
C        RAL COLUMNS OF THE TRIANGULAR ARRAY OF B-SPLINE VALUES ARE NEE-BSPLVB18
C        DED (E.G., IN  BVALUE  OR IN  BSPLVD ). PRECISELY,             BSPLVB19
C                    IF  INDEX = 1 ,                                    BSPLVB20
C        THE CALCULATION STARTS FROM SCRATCH AND THE ENTIRE TRIANGULAR  BSPLVB21
C        ARRAY OF B-SPLINE VALUES OF ORDERS 1,2,...,JHIGH  IS GENERATED BSPLVB22
C        ORDER BY ORDER , I.E., COLUMN BY COLUMN .                      BSPLVB23
C                    IF  INDEX = 2 ,                                    BSPLVB24
C        ONLY THE B-SPLINE VALUES OF ORDER  J+1, J+2, ..., JOUT  ARE GE-BSPLVB25
C        NERATED, THE ASSUMPTION BEING THAT  BIATX , J , DELTAL , DELTARBSPLVB26
C        ARE, ON ENTRY, AS THEY WERE ON EXIT AT THE PREVIOUS CALL.      BSPLVB27
C           IN PARTICULAR, IF  JHIGH = 0, THEN  JOUT = J+1, I.E., JUST  BSPLVB28
C        THE NEXT COLUMN OF B-SPLINE VALUES IS GENERATED.               BSPLVB29
C                                                                       BSPLVB30
C  W A R N I N G . . .  THE RESTRICTION   JOUT .LE. JMAX (= 20)  IS IM- BSPLVB31
C        POSED ARBITRARILY BY THE DIMENSION STATEMENT FOR  DELTAL  AND  BSPLVB32
C        DELTAR  BELOW, BUT IS  N O W H E R E  C H E C K E D  FOR .     BSPLVB33
C                                                                       BSPLVB34
C  X.....THE POINT AT WHICH THE B-SPLINES ARE TO BE EVALUATED.          BSPLVB35
C  LEFT.....AN INTEGER CHOSEN (USUALLY) SO THAT                         BSPLVB36
C                  T(LEFT) .LE. X .LE. T(LEFT+1)  .                     BSPLVB37
C                                                                       BSPLVB38
C******  O U T P U T  ******                                            BSPLVB39
C  BIATX.....ARRAY OF LENGTH  JOUT , WITH  BIATX(I)  CONTAINING THE VAL-BSPLVB40
C        UE AT  X  OF THE POLYNOMIAL OF ORDER  JOUT  WHICH AGREES WITH  BSPLVB41
C        THE B-SPLINE  B(LEFT-JOUT+I,JOUT,T)  ON THE INTERVAL (T(LEFT), BSPLVB42
C        T(LEFT+1)) .                                                   BSPLVB43
C                                                                       BSPLVB44
C******  M E T H O D  ******                                            BSPLVB45
C  THE RECURRENCE RELATION                                              BSPLVB46
C                                                                       BSPLVB47
C                         X - T(I)              T(I+J+1) - X            BSPLVB48
C      B(I,J+1)(X)  =  -----------B(I,J)(X) + ---------------B(I+1,J)(X) BSPLVB49
C                      T(I+J)-T(I)            T(I+J+1)-T(I+1)           BSPLVB50
C                                                                       BSPLVB51
C  IS USED (REPEATEDLY) TO GENERATE THE (J+1)-VECTOR  B(LEFT-J,J+1)(X), BSPLVB52
C  ...,B(LEFT,J+1)(X)  FROM THE J-VECTOR  B(LEFT-J+1,J)(X),...,         BSPLVB53
C  B(LEFT,J)(X), STORING THE NEW VALUES IN  BIATX  OVER THE OLD. THE    BSPLVB54
C  FACTS THAT                                                           BSPLVB55
C            B(I,1) = 1  IF  T(I) .LE. X .LT. T(I+1)                    BSPLVB56
C  AND THAT                                                             BSPLVB57
```

```
C          B(I,J)(X) = 0  UNLESS  T(I) .LE. X .LT. T(I+J)              BSPLVB58
C  ARE USED. THE PARTICULAR ORGANIZATION OF THE CALCULATIONS FOLLOWS AL-BSPLVB59
C  GORITHM  (8)  IN CHAPTER X OF THE TEXT.                              BSPLVB60
C                                                                       BSPLVB61
C      PARAMETER JMAX = 20                                              BSPLVB62
       INTEGER INDEX,JHIGH,LEFT,I,J,JP1                                 BSPLVB63
C      REAL BIATX(JHIGH),T(1),X,   DELTAL(JMAX),DELTAR(JMAX),SAVED,TERM BSPLVB64
       REAL BIATX(JHIGH),T(1),X,DELTAL(20),DELTAR(20),SAVED,TERM        BSPLVB65
C      DIMENSION BIATX(JOUT), T(LEFT+JOUT)                              BSPLVB66
CURRENT FORTRAN STANDARD MAKES IT IMPOSSIBLE TO SPECIFY THE LENGTH OF   BSPLVB67
C  T  AND OF  BIATX  PRECISELY WITHOUT THE INTRODUCTION OF OTHERWISE    BSPLVB68
C  SUPERFLUOUS ADDITIONAL ARGUMENTS.                                    BSPLVB69
       DATA J /1/                                                       BSPLVB70
C      SAVE J,DELTAL,DELTAR (VALID IN FORTRAN 77)                       BSPLVB71
C                                                                       BSPLVB72
       GO TO (10,20), INDEX                                             BSPLVB73
10     J=1                                                              BSPLVB74
       BIATX(1)=1.                                                      BSPLVB75
       IF (J.GE.JHIGH) GO TO 40                                         BSPLVB76
C                                                                       BSPLVB77
20     JP1=J+1                                                          BSPLVB78
       L=LEFT+J                                                         BSPLVB79
       DELTAR(J)=T(L)-X                                                 BSPLVB80
       L=LEFT+1-J                                                       BSPLVB81
       DELTAL(J)=X-T(L)                                                 BSPLVB82
       SAVED=0.                                                         BSPLVB83
       DO 30 I=1,J                                                      BSPLVB84
       L=JP1-I                                                          BSPLVB85
       TERM=BIATX(I)/(DELTAR(I)+DELTAL(L))                              BSPLVB86
       BIATX(I)=SAVED+DELTAR(I)*TERM                                    BSPLVB87
30     SAVED=DELTAL(L)*TERM                                             BSPLVB88
       BIATX(JP1)=SAVED                                                 BSPLVB89
       J=JP1                                                            BSPLVB90
       IF (J.LT.JHIGH) GO TO 20                                         BSPLVB91
C                                                                       BSPLVB92
40     RETURN                                                           BSPLVB93
       END                                                              BSPLVB94
```

CPR*NS(1).BVALUE(2)

```
      REAL FUNCTION BVALUE(T,BCOEF,N,K,X,JDERIV)                           BVALU001
C  FROM  * A PRACTICAL GUIDE TO SPLINES *  BY C. DE BOOR                   BVALU002
CALLS  INTERV                                                              BVALU003
C                                                                          BVALU004
CALCULATES VALUE AT  X  OF  JDERIV-TH DERIVATIVE OF SPLINE FROM B-REPR.  BVALU005
C  THE SPLINE IS TAKEN TO BE CONTINUOUS FROM THE RIGHT.                    BVALU006
C                                                                          BVALU007
C******  I N P U T ******                                                  BVALU008
C  T, BCOEF, N, K......FORMS THE B-REPRESENTATION OF THE SPLINE  F  TO    BVALU009
C        BE EVALUATED. SPECIFICALLY,                                       BVALU010
C  T.....KNOT SEQUENCE, OF LENGTH  N+K, ASSUMED NONDECREASING.             BVALU011
C  BCOEF.....B-COEFFICIENT SEQUENCE, OF LENGTH  N .                        BVALU012
C  N.....LENGTH OF  BCOEF  AND DIMENSION OF SPLINE(K,T),                   BVALU013
C        A S S U M E D  POSITIVE .                                         BVALU014
C  K.....ORDER OF THE SPLINE .                                             BVALU015
C                                                                          BVALU016
C  W A R N I N G . . .   THE RESTRICTION  K .LE. KMAX (=20)  IS IMPOSED BVALU017
C        ARBITRARILY BY THE DIMENSION STATEMENT FOR  AJ, DL, DR  BELOW,  BVALU018
C        BUT IS  N O W H E R E  C H E C K E D  FOR.                        BVALU019
C                                                                          BVALU020
C  X.....THE POINT AT WHICH TO EVALUATE .                                  BVALU021
C  JDERIV.....INTEGER GIVING THE ORDER OF THE DERIVATIVE TO BE EVALUATEDBVALU022
C        A S S U M E D  TO BE ZERO OR POSITIVE.                            BVALU023
C                                                                          BVALU024
C******  O U T P U T  ******                                               BVALU025
C  BVALUE.....THE VALUE OF THE (JDERIV)-TH DERIVATIVE OF  F  AT  X .      BVALU026
C                                                                          BVALU027
C******  M E T H O D  ******                                               BVALU028
C     THE NONTRIVIAL KNOT INTERVAL  (T(I),T(I+1))  CONTAINING  X  IS LO-BVALU029
C  CATED WITH THE AID OF  INTERV . THE  K  B-COEFFS OF  F  RELEVANT FOR BVALU030
C  THIS INTERVAL ARE THEN OBTAINED FROM  BCOEF (OR TAKEN TO BE ZERO IF  BVALU031
C  NOT EXPLICITLY AVAILABLE) AND ARE THEN DIFFERENCED  JDERIV  TIMES TO BVALU032
C  OBTAIN THE B-COEFFS OF  (D**JDERIV)F  RELEVANT FOR THAT INTERVAL.    BVALU033
C  PRECISELY, WITH  J = JDERIV, WE HAVE FROM X.(12) OF THE TEXT THAT     BVALU034
C                                                                          BVALU035
C     (D**J)F  =  SUM ( BCOEF(.,J)*B(.,K-J,T) )                            BVALU036
C                                                                          BVALU037
C  WHERE                                                                   BVALU038
C                   / BCOEF(.),                     ,  J .EQ. 0            BVALU039
C                   /                                                      BVALU040
C    BCOEF(.,J)  =  / BCOEF(.,J-1) - BCOEF(.-1,J-1)                        BVALU041
C                   / ----------------------------- ,  J .GT. 0            BVALU042
C                   /    (T(.+K-J) - T(.))/(K-J)                           BVALU043
C                                                                          BVALU044
C     THEN, WE USE REPEATEDLY THE FACT THAT                                BVALU045
C                                                                          BVALU046
C    SUM ( A(.)*B(.,M,T)(X) )  =  SUM ( A(.,X)*B(.,M-1,T)(X) )             BVALU047
C  WITH                                                                    BVALU048
C                  (X - T(.))*A(.) + (T(.+M-1) - X)*A(.-1)                 BVALU049
C    A(.,X)  =     ---------------------------------------                 BVALU050
C                  (X - T(.))      + (T(.+M-1) - X)                        BVALU051
C                                                                          BVALU052
C  TO WRITE  (D**J)F(X)  EVENTUALLY AS A LINEAR COMBINATION OF B-SPLINESBVALU053
C  OF ORDER  1 , AND THE COEFFICIENT FOR  B(I,1,T)(X)  MUST THEN BE THE BVALU054
C  DESIRED NUMBER  (D**J)F(X). (SEE X.(17)-(19) OF TEXT).                  BVALU055
C                                                                          BVALU056
C     PARAMETER KMAX = 20                                                  BVALU057
```

```
      INTEGER JDERIV,K,N,I,ILO,IMK,J,JC,JCMIN,JCMAX,JJ,KMJ,KM1,MFLAG,NMIBVALU058
     2,JDRVP1                                                           BVALU059
      REAL BCOEF(N),T(1),X,AJ(20),DL(20),DR(20),FKMJ                    BVALU060
C     REAL BCOEF(N),T(1),X,   AJ(KMAX),DL(KMAX),DR(KMAX),FKMJ           BVALU061
C     DIMENSION T(N+K)                                                  BVALU062
CURRENT FORTRAN STANDARD MAKES IT IMPOSSIBLE TO SPECIFY THE LENGTH OF  TBVALU063
C  PRECISELY WITHOUT THE INTRODUCTION OF OTHERWISE SUPERFLUOUS ADDITION-BVALU064
C  AL ARGUMENTS.                                                        BVALU065
      BVALUE=0.                                                         BVALU066
      IF (JDERIV.GE.K) GO TO 170                                        BVALU067
C                                                                       BVALU068
C  *** FIND  I   S.T.   1 .LE. I .LT. N+K   AND   T(I) .LT. T(I+1)   ANDBVALU069
C      T(I) .LE. X .LT. T(I+1) . IF NO SUCH I CAN BE FOUND,  X  LIES    BVALU070
C      OUTSIDE THE SUPPORT OF  THE SPLINE  F  AND  BVALUE = 0.          BVALU071
C      (THE ASYMMETRY IN THIS CHOICE OF  I  MAKES  F  RIGHTCONTINUOUS)  BVALU072
      CALL INTERV (T,N+K,X,I,MFLAG)                                     BVALU073
      IF (MFLAG.NE.0) GO TO 170                                         BVALU074
C  *** IF K = 1 (AND JDERIV = 0), BVALUE = BCOEF(I).                    BVALU075
      KM1=K-1                                                           BVALU076
      IF (KM1.GT.0) GO TO 10                                            BVALU077
      BVALUE=BCOEF(I)                                                   BVALU078
      GO TO 170                                                         BVALU079
C                                                                       BVALU080
C  *** STORE THE K B-SPLINE COEFFICIENTS RELEVANT FOR THE KNOT INTERVAL BVALU081
C     (T(I),T(I+1)) IN AJ(1),...,AJ(K) AND COMPUTE DL(J) = X - T(I+1-J),BVALU082
C     DR(J) = T(I+J) - X, J=1,...,K-1 . SET ANY OF THE AJ NOT OBTAINABLEBVALU083
C     FROM INPUT TO ZERO. SET ANY T.S NOT OBTAINABLE EQUAL TO T(1) OR   BVALU084
C     TO T(N+K) APPROPRIATELY.                                          BVALU085
10    JCMIN=1                                                           BVALU086
      IMK=I-K                                                           BVALU087
      IF (IMK.GE.0) GO TO 40                                            BVALU088
      JCMIN=1-IMK                                                       BVALU089
      DO 20 J=1,I                                                       BVALU090
      L=I+1-J                                                           BVALU091
20    DL(J)=X-T(L)                                                      BVALU092
      DO 30 J=I,KM1                                                     BVALU093
      L=K-J                                                             BVALU094
      AJ(L)=0.                                                          BVALU095
30    DL(J)=DL(I)                                                       BVALU096
      GO TO 60                                                          BVALU097
40    DO 50 J=1,KM1                                                     BVALU098
      L=I+1-J                                                           BVALU099
50    DL(J)=X-T(L)                                                      BVALU100
C                                                                       BVALU101
60    JCMAX=K                                                           BVALU102
      NMI=N-I                                                           BVALU103
      IF (NMI.GE.0) GO TO 90                                            BVALU104
      JCMAX=K+NMI                                                       BVALU105
      DO 70 J=1,JCMAX                                                   BVALU106
      L=I+J                                                             BVALU107
70    DR(J)=T(L)-X                                                      BVALU108
      DO 80 J=JCMAX,KM1                                                 BVALU109
      AJ(J+1)=0.                                                        BVALU110
80    DR(J)=DR(JCMAX)                                                   BVALU111
      GO TO 110                                                         BVALU112
90    DO 100 J=1,KM1                                                    BVALU113
      L=I+J                                                             BVALU114
100   DR(J)=T(L)-X                                                      BVALU115
```

```
C                                                                       BVALU116
  110   DO 120 JC=JCMIN,JCMAX                                           BVALU117
        L=IMK+JC                                                        BVALU118
  120   AJ(JC)=BCOEF(L)                                                 BVALU119
C                                                                       BVALU120
C                 *** DIFFERENCE THE COEFFICIENTS  JDERIV  TIMES.       BVALU121
        IF (JDERIV.EQ.0) GO TO 140                                      BVALU122
        DO 130 J=1,JDERIV                                               BVALU123
        KMJ=K-J                                                         BVALU124
        FKMJ=FLOAT(KMJ)                                                 BVALU125
        ILO=KMJ                                                         BVALU126
        DO 130 JJ=1,KMJ                                                 BVALU127
        AJ(JJ)=((AJ(JJ+1)-AJ(JJ))/(DL(ILO)+DR(JJ)))*FKMJ                BVALU128
  130   ILO=ILO-1                                                       BVALU129
C                                                                       BVALU130
C  *** COMPUTE VALUE AT  X  IN (T(I),T(I+1)) OF JDERIV-TH DERIVATIVE,   BVALU131
C     GIVEN ITS RELEVANT B-SPLINE COEFFS IN AJ(1),...,AJ(K-JDERIV).     BVALU132
  140   IF (JDERIV.EQ.KM1) GO TO 160                                    BVALU133
        JDRVP1=JDERIV+1                                                 BVALU134
        DO 150 J=JDRVP1,KM1                                             BVALU135
        KMJ=K-J                                                         BVALU136
        ILO=KMJ                                                         BVALU137
        DO 150 JJ=1,KMJ                                                 BVALU138
        AJ(JJ)=(AJ(JJ+1)*DL(ILO)+AJ(JJ)*DR(JJ))/(DL(ILO)+DR(JJ))        BVALU139
  150   ILO=ILO-1                                                       BVALU140
  160   BVALUE=AJ(1)                                                    BVALU141
C                                                                       BVALU142
  170   RETURN                                                          BVALU143
        END                                                             BVALU144
```

```
CPR*NS(1).CHECK1(2)
      SUBROUTINE CHECK1 (W,N,NX,K,KX,NKX,NY,NYX,JX,MO,AL,DL,C,NZ)       CHECK101
C-----------------------------------------------------------------------CHECK102
C     CHECK1   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING     CHECK103
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. CHECK104
C              AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION CHECK105
C     FOR: CHECKING WHETHER INPUT VALUES FALL WITHIN THEIR ALLOWABLE    CHECK106
C          LIMITS                                                       CHECK107
C     SUBPROGRAMS CALLED: -NONE-                                        CHECK108
C     CURRENT VERSION COMPLETED JUNE 20, 1980                           CHECK109
C-----------------------------------------------------------------------CHECK110
      DIMENSION W(NX)                                                   CHECK111
C--- WRITE FORMATS                                                      CHECK112
10    FORMAT (/1X,21H*** VECTOR LENGTH N =,I4,2X,20HEXCEEDS DIMENSIONED CHECK113
     2,10HVALUE NX =,I4)                                                CHECK114
20    FORMAT (/1X,18H*** DIMENSION KX =,I4,2X,20HMUST BE AT LEAST AS ,  CHECK115
     2 8HLARGE AS/5X,26HK + 2*(DEGREE OF SPLINE) =,I4)                  CHECK116
30    FORMAT (/1X,22H*** VECTOR LENGTH NY =,I4,2X,                      CHECK117
     2 20HEXCEEDS DIMENSIONED ,11HVALUE NYX =,I4)                       CHECK118
40    FORMAT (/1X,17H*** WEIGHT NUMBER,I5,1X,13HIS NEGATIVE (,G10.5,1H))CHECK119
50    FORMAT (/1X,22H*** DEGREE OF SPLINE (,I3,12H) EXCEEDS 19)         CHECK120
60    FORMAT (/1X,28H*** NUMBER OF OBSERVATIONS (,I4,13H) MUST EXCEED,  CHECK121
     2 I5)                                                              CHECK122
70    FORMAT (/1X,33H*** ALPHA LEVEL OF SIGNIFICANCE (,F6.3,            CHECK123
     2 10H) MUST BE ,21HIN THE INTERVAL (0,1])                          CHECK124
80    FORMAT (/1X,33H*** DELTA LEVEL OF SIGNIFICANCE (,F6.3,            CHECK125
     2 10H) MUST BE ,21HIN THE INTERVAL (0,1])                          CHECK126
90    FORMAT (/1X,16H*** CONSTANT C (,F6.3,26H) MUST BE IN THE INTERVAL CHECK127
     2,11H[0.85,1.25])                                                  CHECK128
100   FORMAT (//1X,I4,1X,40HERROR CONDITIONS DETECTED BY SUBROUTINE ,   CHECK129
     2 8H*CHECK1*//6X,38H*****PROGRAM EXECUTION TERMINATED*****//)      CHECK130
110   FORMAT (/1X,47H*** MAXIMUM ORDER OF SPLINES JX MUST BE 20 (NOT,I3,CHECK131
     2 1H))                                                             CHECK132
120   FORMAT (/5X,42HSEE APPENDIX 1 OF THE FOLLOWING NBS PAPER://5X,    CHECK133
     2 36HA NEW APPROACH TO VOLUME CALIBRATION/5X,                      CHECK134
     3 51HBY J. A. LECHNER, C. P. REEVE, AND C. H. SPIEGELMAN/)         CHECK135
130   FORMAT (/1X,23H*** VECTOR LENGTH N+K =,I4,2X,                     CHECK136
     2 20HEXCEEDS DIMENSIONED ,11HVALUE NKX =,I4)                       CHECK137
C--- INITIALIZE COUNT FOR ERROR CONDITIONS                              CHECK138
      KOUNT=0                                                           CHECK139
C--- INITIALIZE NUMBER OF ZERO WEIGHTS                                  CHECK140
      NZ=0                                                              CHECK141
C--- CHECK FOR VECTOR LENGTHS EXCEEDING DIMENSIONED VALUES              CHECK142
      IF (N.LE.NX) GO TO 140                                            CHECK143
      KOUNT=KOUNT+1                                                     CHECK144
      WRITE (6,10) N,NX                                                 CHECK145
140   NK=N+K                                                            CHECK146
      IF (NK.LE.NKX) GO TO 150                                          CHECK147
      KOUNT=KOUNT+1                                                     CHECK148
      WRITE (6,130) NK,NKX                                              CHECK149
150   K2=K+2*(MO-1)                                                     CHECK150
      IF (K2.LE.KX) GO TO 160                                           CHECK151
      KOUNT=KOUNT+1                                                     CHECK152
      WRITE (6,20) KX,K2                                                CHECK153
160   IF (NY.LE.NYX) GO TO 170                                          CHECK154
      KOUNT=KOUNT+1                                                     CHECK155
      WRITE (6,30) NY,NYX                                               CHECK156
C--- CHECK FOR NEGATIVE AND ZERO WEIGHTS                                CHECK157
```

```
170   DO 200 I=1,N                                                      CHECK158
      IF (W(I)) 180,190,200                                             CHECK159
C---  COUNT EACH NEGATIVE WEIGHT AS AN ERROR CONDITION                  CHECK160
180   KOUNT=KOUNT+1                                                     CHECK161
      WRITE (6,40) I,W(I)                                               CHECK162
      GO TO 200                                                         CHECK163
C---  COUNT ZERO WEIGHTS                                                CHECK164
190   NZ=NZ+1                                                           CHECK165
200   CONTINUE                                                          CHECK166
C---  CHECK FOR MAXIMUM ORDER OF SPLINE = 20                            CHECK167
      IF (JX.EQ.20) GO TO 210                                           CHECK168
      KOUNT=KOUNT+1                                                     CHECK169
      WRITE (6,110) JX                                                  CHECK170
C---  CHECK ORDER OF SPLINE                                             CHECK171
210   IF (MO.LE.20) GO TO 220                                           CHECK172
      KOUNT=KOUNT+1                                                     CHECK173
      MD=MO-1                                                           CHECK174
      WRITE (6,50) MD                                                   CHECK175
C---  CHECK NUMBER OF OBSERVATIONS                                      CHECK176
220   K2=K+MO-2+NZ                                                      CHECK177
      IF (N.GT.K2) GO TO 230                                            CHECK178
      KOUNT=KOUNT+1                                                     CHECK179
      WRITE (6,60) N,K2                                                 CHECK180
C---  CHECK SIGNIFICANCE LEVELS                                         CHECK181
230   IF (AL.GT.0.0.AND.AL.LE.1.0) GO TO 240                            CHECK182
      KOUNT=KOUNT+1                                                     CHECK183
      WRITE (6,70) AL                                                   CHECK184
240   IF (DL.GT.0.0.AND.DL.LE.1.0) GO TO 250                            CHECK185
      KOUNT=KOUNT+1                                                     CHECK186
      WRITE (6,80) DL                                                   CHECK187
C---  CHECK CONSTANT C                                                  CHECK188
250   IF (C.LE.1.25.AND.C.GE.0.85) GO TO 260                            CHECK189
      KOUNT=KOUNT+1                                                     CHECK190
      WRITE (6,90) C                                                    CHECK191
      WRITE (6,120)                                                     CHECK192
C---  CHECK WHETHER ANY ERROR CONDITIONS EXIST                          CHECK193
260   IF (KOUNT.EQ.0) RETURN                                            CHECK194
      WRITE (6,100) KOUNT                                               CHECK195
      STOP                                                              CHECK196
      END                                                               CHECK197
```

CPR*NS(1).CHECK2(1)

```
      SUBROUTINE CHECK2 (T,K,KX,X,W,N,NX,NZ,MO)                          CHECK201
C-----------------------------------------------------------------------CHECK202
C    CHECK2   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING       CHECK203
C             DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.   CHECK204
C             AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION   CHECK205
C    FOR: CHECKING FOR OBSERVATIONS WHICH LIE OUTSIDE THE SEQUENCE OF    CHECK206
C         KNOTS.  THE WEIGHTS OF SUCH OBSERVATIONS ARE SET TO ZERO.      CHECK207
C    SUBPROGRAMS CALLED: -NONE-                                          CHECK208
C    CURRENT VERSION COMPLETED MARCH 24, 1980                            CHECK209
C-----------------------------------------------------------------------CHECK210
      DIMENSION T(KX),X(NX),W(NX)                                        CHECK211
   10 FORMAT (/1X,6H*** X(,I4,3H) =,G12.6,1X,22HIS OUTSIDE KNOT SPAN. , CHECK212
     2 1X,11HSET WEIGHT(,I4,5H) =0.)                                     CHECK213
   20 FORMAT (/1X,48H*** ADDITIONAL ZERO WEIGHTS GIVE NONPOSITIVE ***/9XCHECK214
     2,32HDEGREES OF FREEDOM FOR RESIDUALS//6X,13H*****PROGRAM ,         CHECK215
     3 25HEXECUTION TERMINATED*****//)                                   CHECK216
   30 FORMAT (/1X,15H*** VALUE OF X(,I4,14H) CHANGED FROM,G14.8,2X,2HTO,CHECK217
     2 G14.8/5X,45HSO THAT IT WILL BE LESS THAN THE LARGEST KNOT)        CHECK218
      DO 60 I=1,N                                                        CHECK219
      IF (W(I).EQ.0.0) GO TO 60                                          CHECK220
      IF (X(I).LT.T(1)) GO TO 50                                         CHECK221
      IF (X(I)-T(K)) 60,40,50                                            CHECK222
   40 XOLD=X(I)                                                          CHECK223
      X(I)=XOLD-ABS(XOLD)*0.0000001                                      CHECK224
      WRITE (6,30) I,XOLD,X(I)                                           CHECK225
      GO TO 60                                                           CHECK226
   50 W(I)=0.0                                                           CHECK227
      NZ=NZ+1                                                            CHECK228
      WRITE (6,10) I,X(I),I                                              CHECK229
   60 CONTINUE                                                           CHECK230
      K2=K+MO-2+NZ                                                       CHECK231
      IF (N.GT.K2) RETURN                                                CHECK232
      WRITE (6,20)                                                       CHECK233
      STOP                                                               CHECK234
      END                                                                CHECK235
```

```
CPR*NS(1).CHSCDF(1)
      SUBROUTINE CHSCDF (X,NU,CDF)                                      CHSCD001
C                                                                       CHSCD002
C     PURPOSE--THIS SUBROUTINE COMPUTES THE CUMULATIVE DISTRIBUTION     CHSCD003
C              FUNCTION VALUE FOR THE CHI-SQUARED DISTRIBUTION          CHSCD004
C              WITH INTEGER DEGREES OF FREEDOM PARAMETER = NU.          CHSCD005
C              THIS DISTRIBUTION IS DEFINED FOR ALL NON-NEGATIVE X.     CHSCD006
C              THE PROBABILITY DENSITY FUNCTION IS GIVEN                CHSCD007
C              IN THE REFERENCES BELOW.                                 CHSCD008
C     INPUT  ARGUMENTS--X      = THE SINGLE PRECISION VALUE AT          CHSCD009
C                                WHICH THE CUMULATIVE DISTRIBUTION      CHSCD010
C                                FUNCTION IS TO BE EVALUATED.           CHSCD011
C                                X SHOULD BE NON-NEGATIVE.              CHSCD012
C                      --NU    = THE INTEGER NUMBER OF DEGREES          CHSCD013
C                                OF FREEDOM.                            CHSCD014
C                                NU SHOULD BE POSITIVE.                 CHSCD015
C     OUTPUT ARGUMENTS--CDF    = THE SINGLE PRECISION CUMULATIVE        CHSCD016
C                                DISTRIBUTION FUNCTION VALUE.           CHSCD017
C     OUTPUT--THE SINGLE PRECISION CUMULATIVE DISTRIBUTION              CHSCD018
C             FUNCTION VALUE CDF FOR THE CHI-SQUARED DISTRIBUTION       CHSCD019
C             WITH DEGREES OF FREEDOM PARAMETER = NU.                   CHSCD020
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.   CHSCD021
C     RESTRICTIONS--X SHOULD BE NON-NEGATIVE.                           CHSCD022
C                 --NU SHOULD BE A POSITIVE INTEGER VARIABLE.           CHSCD023
C     OTHER DATAPAC   SUBROUTINES NEEDED--NORCDF.                       CHSCD024
C     FORTRAN LIBRARY SUBROUTINES NEEDED--DSQRT, DEXP.                  CHSCD025
C     MODE OF INTERNAL OPERATIONS--DOUBLE PRECISION.                    CHSCD026
C     LANGUAGE--ANSI FORTRAN.                                           CHSCD027
C     REFERENCES--NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS      CHSCD028
C                 SERIES 55, 1964, PAGE 941, FORMULAE 26.4.4 AND 26.4.5.CHSCD029
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE               CHSCD030
C                 DISTRIBUTIONS--1, 1970, PAGE 176,                     CHSCD031
C                 FORMULA 28, AND PAGE 180, FORMULA 33.1.               CHSCD032
C               --OWEN, HANDBOOK OF STATISTICAL TABLES,                 CHSCD033
C                 1962, PAGES 50-55.                                    CHSCD034
C               --PEARSON AND HARTLEY, BIOMETRIKA TABLES                CHSCD035
C                 FOR STATISTICIANS, VOLUME 1, 1954,                    CHSCD036
C                 PAGES 122-131.                                        CHSCD037
C     WRITTEN BY--JAMES J. FILLIBEN                                     CHSCD038
C                 STATISTICAL ENGINEERING LABORATORY (205.03)           CHSCD039
C                 NATIONAL BUREAU OF STANDARDS                          CHSCD040
C                 WASHINGTON, D. C. 20234                               CHSCD041
C                 PHONE:  301-921-2315                                  CHSCD042
C     ORIGINAL VERSION--JUNE      1972.                                 CHSCD043
C     UPDATED         --MAY       1974.                                 CHSCD044
C     UPDATED         --SEPTEMBER 1975.                                 CHSCD045
C     UPDATED         --NOVEMBER  1975.                                 CHSCD046
C     UPDATED         --OCTOBER   1976.                                 CHSCD047
C                                                                       CHSCD048
C-----------------------------------------------------------------      CHSCD049
C                                                                       CHSCD050
      DOUBLE PRECISION DX,PI,CHI,SUM,TERM,AI,DCDFN                      CHSCD051
      DOUBLE PRECISION DNU                                              CHSCD052
      DOUBLE PRECISION DSQRT,DEXP                                       CHSCD053
      DOUBLE PRECISION DLOG                                             CHSCD054
      DOUBLE PRECISION DFACT,DPOWER                                     CHSCD055
      DOUBLE PRECISION DW                                               CHSCD056
      DOUBLE PRECISION D1,D2,D3                                         CHSCD057
```

```
      DOUBLE PRECISION TERM0,TERM1,TERM2,TERM3,TERM4                    CHSCD058
      DOUBLE PRECISION B11                                              CHSCD059
      DOUBLE PRECISION B21                                              CHSCD060
      DOUBLE PRECISION B31,B32                                          CHSCD061
      DOUBLE PRECISION B41,B42,B43                                      CHSCD062
      DATA NUCUT /1000/                                                 CHSCD063
      DATA PI /3.14159265358979D0/                                      CHSCD064
      DATA DPOWER /0.33333333333333D0/                                  CHSCD065
      DATA B11 /0.33333333333333D0/                                     CHSCD066
      DATA B21 /-0.02777777777778D0/                                    CHSCD067
      DATA B31 /-0.00061728395061D0/                                    CHSCD068
      DATA B32 /-13.0D0/                                                CHSCD069
      DATA B41 /0.00018004115226D0/                                     CHSCD070
      DATA B42 /6.0D0/                                                  CHSCD071
      DATA B43 /17.0D0/                                                 CHSCD072
C                                                                       CHSCD073
      IPR=6                                                             CHSCD074
C                                                                       CHSCD075
C     CHECK THE INPUT ARGUMENTS FOR ERRORS                              CHSCD076
C                                                                       CHSCD077
      IF (NU.LE.0) GO TO 10                                             CHSCD078
      IF (X.LT.0.0) GO TO 20                                            CHSCD079
      GO TO 30                                                          CHSCD080
   10 WRITE (IPR,50)                                                    CHSCD081
      WRITE (IPR,70) NU                                                 CHSCD082
      CDF=0.0                                                           CHSCD083
      RETURN                                                            CHSCD084
   20 WRITE (IPR,40)                                                    CHSCD085
      WRITE (IPR,60) X                                                  CHSCD086
      CDF=0.0                                                           CHSCD087
      RETURN                                                            CHSCD088
   30 CONTINUE                                                          CHSCD089
   40 FORMAT (1H ,96H***** NON-FATAL DIAGNOSTIC--THE FIRST  INPUT ARGUMECHSCD090
     2NT TO THE CHSCDF SUBROUTINE IS NEGATIVE *****)                    CHSCD091
   50 FORMAT (1H ,91H***** FATAL ERROR--THE SECOND INPUT ARGUMENT TO THECHSCD092
     2 CHSCDF SUBROUTINE IS NON-POSITIVE *****)                         CHSCD093
   60 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)CHSCD094
   70 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,I8,6H *****)   CHSCD095
C                                                                       CHSCD096
C-----START POINT-----------------------------------------------------  CHSCD097
C                                                                       CHSCD098
      DX=X                                                              CHSCD099
      ANU=NU                                                            CHSCD100
      DNU=NU                                                            CHSCD101
C                                                                       CHSCD102
C     IF X IS NON-POSITIVE, SET CDF = 0.0 AND RETURN.                   CHSCD103
C     IF NU IS SMALLER THAN 10 AND X IS MORE THAN 200                   CHSCD104
C     STANDARD DEVIATIONS BELOW THE MEAN,                               CHSCD105
C     SET CDF = 0.0 AND RETURN.                                         CHSCD106
C     IF NU IS 10 OR LARGER AND X IS MORE THAN 100                      CHSCD107
C     STANDARD DEVIATIONS BELOW THE MEAN,                               CHSCD108
C     SET CDF = 0.0 AND RETURN.                                         CHSCD109
C     IF NU IS SMALLER THAN 10 AND X IS MORE THAN 200                   CHSCD110
C     STANDARD DEVIATIONS ABOVE THE MEAN,                               CHSCD111
C     SET CDF = 1.0 AND RETURN.                                         CHSCD112
C     IF NU IS 10 OR LARGER AND X IS MORE THAN 100                      CHSCD113
C     STANDARD DEVIATIONS ABOVE THE MEAN,                               CHSCD114
C     SET CDF = 1.0 AND RETURN.                                         CHSCD115
```

```
C                                                                       CHSCD116
      IF (X.LE.0.0) GO TO 80                                            CHSCD117
      AMEAN=ANU                                                         CHSCD118
      SD=SQRT(2.0*ANU)                                                  CHSCD119
      Z=(X-AMEAN)/SD                                                    CHSCD120
      IF (NU.LT.10.AND.Z.LT.-200.0) GO TO 80                            CHSCD121
      IF (NU.GE.10.AND.Z.LT.-100.0) GO TO 80                            CHSCD122
      IF (NU.LT.10.AND.Z.GT.200.0) GO TO 90                             CHSCD123
      IF (NU.GE.10.AND.Z.GT.100.0) GO TO 90                             CHSCD124
      GO TO 100                                                         CHSCD125
   80 CDF=0.0                                                           CHSCD126
      RETURN                                                            CHSCD127
   90 CDF=1.0                                                           CHSCD128
      RETURN                                                            CHSCD129
  100 CONTINUE                                                          CHSCD130
C                                                                       CHSCD131
C     DISTINGUISH BETWEEN 3 SEPARATE REGIONS                            CHSCD132
C     OF THE (X,NU) SPACE.                                              CHSCD133
C     BRANCH TO THE PROPER COMPUTATIONAL METHOD                         CHSCD134
C     DEPENDING ON THE REGION.                                          CHSCD135
C     NUCUT HAS THE VALUE 1000.                                         CHSCD136
C                                                                       CHSCD137
      IF (NU.LT.NUCUT) GO TO 120                                        CHSCD138
      IF (NU.GE.NUCUT.AND.X.LE.ANU) GO TO 180                           CHSCD139
      IF (NU.GE.NUCUT.AND.X.GT.ANU) GO TO 190                           CHSCD140
      IBRAN=1                                                           CHSCD141
      WRITE (IPR,110) IBRAN                                             CHSCD142
  110 FORMAT (1H ,42H*****INTERNAL ERROR IN CHSCDF SUBROUTINE--,        CHSCD143
     2 46HIMPOSSIBLE BRANCH CONDITION AT BRANCH POINT = ,I8)            CHSCD144
      RETURN                                                            CHSCD145
C                                                                       CHSCD146
C     TREAT THE SMALL AND MODERATE DEGREES OF FREEDOM CASE              CHSCD147
C     (THAT IS, WHEN NU IS SMALLER THAN 1000).                          CHSCD148
C     METHOD UTILIZED--EXACT FINITE SUM                                 CHSCD149
C     (SEE AMS 55, PAGE 941, FORMULAE 26.4.4 AND 26.4.5).               CHSCD150
C                                                                       CHSCD151
  120 CONTINUE                                                          CHSCD152
      CHI=DSQRT(DX)                                                     CHSCD153
      IEVODD=NU-2*(NU/2)                                                CHSCD154
      IF (IEVODD.EQ.0) GO TO 130                                        CHSCD155
C                                                                       CHSCD156
      SUM=0.0D0                                                         CHSCD157
      TERM=1.0/CHI                                                      CHSCD158
      IMIN=1                                                            CHSCD159
      IMAX=NU-1                                                         CHSCD160
      GO TO 140                                                         CHSCD161
C                                                                       CHSCD162
  130 SUM=1.0D0                                                         CHSCD163
      TERM=1.0D0                                                        CHSCD164
      IMIN=2                                                            CHSCD165
      IMAX=NU-2                                                         CHSCD166
C                                                                       CHSCD167
  140 IF (IMIN.GT.IMAX) GO TO 160                                       CHSCD168
      DO 150 I=IMIN,IMAX,2                                              CHSCD169
      AI=I                                                              CHSCD170
      TERM=TERM*(DX/AI)                                                 CHSCD171
      SUM=SUM+TERM                                                      CHSCD172
  150 CONTINUE                                                          CHSCD173
```

```
  160   CONTINUE                                                        CHSCD174
  C                                                                     CHSCD175
        SUM=SUM*DEXP(-DX/2.0D0)                                         CHSCD176
        IF (IEVODD.EQ.0) GO TO 170                                      CHSCD177
        SUM=(DSQRT(2.0D0/PI))*SUM                                       CHSCD178
        SPCHI=CHI                                                       CHSCD179
        CALL NORCDF (SPCHI,CDFN)                                        CHSCD180
        DCDFN=CDFN                                                      CHSCD181
        SUM=SUM+2.0D0*(1.0D0-DCDFN)                                     CHSCD182
  170   CDF=1.0D0-SUM                                                   CHSCD183
        RETURN                                                          CHSCD184
  C                                                                     CHSCD185
  C     TREAT THE CASE WHEN NU IS LARGE                                 CHSCD186
  C     (THAT IS, WHEN NU IS EQUAL TO OR GREATER THAN 1000)             CHSCD187
  C     AND X IS LESS THAN OR EQUAL TO NU.                              CHSCD188
  C     METHOD UTILIZED--WILSON-HILFERTY APPROXIMATION                  CHSCD189
  C     (SEE JOHNSON AND KOTZ, VOLUME 1, PAGE 176, FORMULA 28).         CHSCD190
  C                                                                     CHSCD191
  180   CONTINUE                                                        CHSCD192
        DFACT=4.5D0*DNU                                                 CHSCD193
        U=(((DX/DNU)**DPOWER)-1.0D0+(1.0D0/DFACT))*DSQRT(DFACT)         CHSCD194
        CALL NORCDF (U,CDFN)                                            CHSCD195
        CDF=CDFN                                                        CHSCD196
        RETURN                                                          CHSCD197
  C                                                                     CHSCD198
  C     TREAT THE CASE WHEN NU IS LARGE                                 CHSCD199
  C     (THAT IS, WHEN NU IS EQUAL TO OR GREATER THAN 1000)             CHSCD200
  C     AND X IS LARGER THAN NU.                                        CHSCD201
  C     METHOD UTILIZED--HILL'S ASYMPTOTIC EXPANSION                    CHSCD202
  C     (SEE JOHNSON AND KOTZ, VOLUME 1, PAGE 180, FORMULA 33.1).       CHSCD203
  C                                                                     CHSCD204
  190   CONTINUE                                                        CHSCD205
        DW=DSQRT(DX-DNU-DNU*DLOG(DX/DNU))                               CHSCD206
        DANU=DSQRT(2.0D0/DNU)                                           CHSCD207
        D1=DW                                                           CHSCD208
        D2=DW**2                                                        CHSCD209
        D3=DW**3                                                        CHSCD210
        TERM0=DW                                                        CHSCD211
        TERM1=B11*DANU                                                  CHSCD212
        TERM2=B21*D1*(DANU**2)                                          CHSCD213
        TERM3=B31*(D2+B32)*(DANU**3)                                    CHSCD214
        TERM4=B41*(B42*D3+B43*D1)*(DANU**4)                             CHSCD215
        U=TERM0+TERM1+TERM2+TERM3+TERM4                                 CHSCD216
        CALL NORCDF (U,CDFN)                                            CHSCD217
        CDF=CDFN                                                        CHSCD218
        RETURN                                                          CHSCD219
  C                                                                     CHSCD220
        END                                                             CHSCD221
```

CPR*NS(1).CHSPPF(1)

```
      SUBROUTINE CHSPPF (P,NU,PPF)                                      CHSPP001
C                                                                       CHSPP002
C     PURPOSE--THIS SUBROUTINE COMPUTES THE PERCENT POINT               CHSPP003
C              FUNCTION VALUE FOR THE CHI-SQUARED DISTRIBUTION          CHSPP004
C              WITH INTEGER DEGREES OF FREEDOM PARAMETER = NU.          CHSPP005
C              THE CHI-SQUARED DISTRIBUTION USED                        CHSPP006
C              HEREIN IS DEFINED FOR ALL NON-NEGATIVE X,                CHSPP007
C              AND ITS PROBABILITY DENSITY FUNCTION IS GIVEN            CHSPP008
C              IN REFERENCES 2, 3, AND 4 BELOW.                         CHSPP009
C              NOTE THAT THE PERCENT POINT FUNCTION OF A DISTRIBUTION   CHSPP010
C              IS IDENTICALLY THE SAME AS THE INVERSE CUMULATIVE        CHSPP011
C              DISTRIBUTION FUNCTION OF THE DISTRIBUTION.               CHSPP012
C     INPUT  ARGUMENTS--P      = THE SINGLE PRECISION VALUE             CHSPP013
C                                (BETWEEN 0.0 (INCLUSIVELY)             CHSPP014
C                                AND 1.0 (EXCLUSIVELY))                 CHSPP015
C                                AT WHICH THE PERCENT POINT             CHSPP016
C                                FUNCTION IS TO BE EVALUATED.           CHSPP017
C                     --NU     = THE INTEGER NUMBER OF DEGREES          CHSPP018
C                                OF FREEDOM.                            CHSPP019
C                                NU SHOULD BE POSITIVE.                 CHSPP020
C     OUTPUT ARGUMENTS--PPF    = THE SINGLE PRECISION PERCENT           CHSPP021
C                                POINT FUNCTION VALUE.                  CHSPP022
C     OUTPUT--THE SINGLE PRECISION PERCENT POINT FUNCTION .             CHSPP023
C             VALUE PPF FOR THE CHI-SQUARED DISTRIBUTION                CHSPP024
C             WITH DEGREES OF FREEDOM PARAMETER = NU.                   CHSPP025
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.   CHSPP026
C     RESTRICTIONS--NU SHOULD BE A POSITIVE INTEGER VARIABLE.           CHSPP027
C                 --P SHOULD BE BETWEEN 0.0 (INCLUSIVELY)               CHSPP028
C                   AND 1.0 (EXCLUSIVELY).                              CHSPP029
C     OTHER DATAPAC   SUBROUTINES NEEDED--NONE.                         CHSPP030
C     FORTRAN LIBRARY SUBROUTINES NEEDED--DEXP, DLOG.                   CHSPP031
C     MODE OF INTERNAL OPERATIONS--DOUBLE PRECISION.                    CHSPP032
C     LANGUAGE--ANSI FORTRAN.                                           CHSPP033
C     ACCURACY--(ON THE UNIVAC 1108, EXEC 8 SYSTEM AT NBS)              CHSPP034
C               COMPARED TO THE KNOWN NU = 2 (EXPONENTIAL)              CHSPP035
C               RESULTS, AGREEMENT WAS HAD OUT TO 6 SIGNIFICANT         CHSPP036
C               DIGITS FOR ALL TESTED P IN THE RANGE P = .001 TO        CHSPP037
C               P = .999.  FOR P = .95 AND SMALLER, THE AGREEMENT       CHSPP038
C               WAS EVEN BETTER--7 SIGNIFICANT DIGITS.                  CHSPP039
C               (NOTE THAT THE TABULATED VALUES GIVEN IN THE WILK,      CHSPP040
C               GNANADESIKAN, AND HUYETT REFERENCE BELOW, PAGE 20,      CHSPP041
C               ARE IN ERROR FOR AT LEAST THE GAMMA = 1 CASE--          CHSPP042
C               THE WORST DETECTED ERROR WAS AGREEMENT TO ONLY 3        CHSPP043
C               SIGNIFICANT DIGITS (IN THEIR 8 SIGNIFICANT DIGIT TABLE) CHSPP044
C               FOR P = .999.)                                          CHSPP045
C     REFERENCES--WILK, GNANADESIKAN, AND HUYETT, 'PROBABILITY          CHSPP046
C                 PLOTS FOR THE GAMMA DISTRIBUTION',                    CHSPP047
C                 TECHNOMETRICS, 1962, PAGES 1-15,                      CHSPP048
C                 ESPECIALLY PAGES 3-5.                                 CHSPP049
C               --NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS      CHSPP050
C                 SERIES 55, 1964, PAGE 257, FORMULA 6.1.41,            CHSPP051
C                 AND PAGES 940-943.                                    CHSPP052
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE               CHSPP053
C                 DISTRIBUTIONS--1, 1970, PAGES 166-206.                CHSPP054
C               --HASTINGS AND PEACOCK, STATISTICAL                     CHSPP055
C                 DISTRIBUTIONS--A HANDBOOK FOR                         CHSPP056
C                 STUDENTS AND PRACTITIONERS, 1975,                     CHSPP057
```

```
C               PAGES 46-51.                                             CHSPP058
C     WRITTEN BY--JAMES J. FILLIBEN                                      CHSPP059
C                 STATISTICAL ENGINEERING LABORATORY (205.03)           CHSPP060
C                 NATIONAL BUREAU OF STANDARDS                          CHSPP061
C                 WASHINGTON, D. C. 20234                               CHSPP062
C                 PHONE:  301-921-2315                                  CHSPP063
C     ORIGINAL VERSION--SEPTEMBER 1975.                                 CHSPP064
C     UPDATED         --NOVEMBER  1975.                                 CHSPP065
C                                                                        CHSPP066
C-----------------------------------------------------------------------CHSPP067
C                                                                        CHSPP068
      DOUBLE PRECISION DP,DGAMMA                                         CHSPP069
      DOUBLE PRECISION Z,Z2,Z3,Z4,Z5,DEN,A,B,C,D,G                      CHSPP070
      DOUBLE PRECISION XMIN0,XMIN,AI,XMAX,DX,PCALC,XMID                 CHSPP071
      DOUBLE PRECISION XLOWER,XUPPER,XDEL                                CHSPP072
      DOUBLE PRECISION SUM,TERM,CUT1,CUT2,AJ,CUTOFF,T                   CHSPP073
      DOUBLE PRECISION DEXP,DLOG                                         CHSPP074
      DIMENSION D(10)                                                    CHSPP075
      DATA C /.918938533204672741D0/                                     CHSPP076
      DATA D(1),D(2),D(3),D(4),D(5) /+.833333333333333333D-1,           CHSPP077
     2 -.277777777777777778D-2,+.793650793650793651D-3,                 CHSPP078
     3 -.595238095238095238D-3,+.841750841750841751D-3/                 CHSPP079
      DATA D(6),D(7),D(8),D(9),D(10) /-.191752691752691753D-2,          CHSPP080
     2 +.641025641025641025D-2,-.295506535947712418D-1,                 CHSPP081
     3 +.179644372368830573D0,-.139243221690590111D1/                   CHSPP082
C                                                                        CHSPP083
      IPR=6                                                              CHSPP084
C                                                                        CHSPP085
C     CHECK THE INPUT ARGUMENTS FOR ERRORS                               CHSPP086
C                                                                        CHSPP087
      IF (P.LT.0.0.OR.P.GE.1.0) GO TO 10                                 CHSPP088
      IF (NU.LT.1) GO TO 20                                              CHSPP089
      GO TO 30                                                           CHSPP090
   10 WRITE (IPR,40)                                                     CHSPP091
      WRITE (IPR,60) P                                                   CHSPP092
      PPF=0.0                                                            CHSPP093
      RETURN                                                             CHSPP094
   20 WRITE (IPR,50)                                                     CHSPP095
      WRITE (IPR,70) NU                                                  CHSPP096
      PPF=0.0                                                            CHSPP097
      RETURN                                                             CHSPP098
   30 CONTINUE                                                           CHSPP099
   40 FORMAT (1H ,115H***** FATAL ERROR--THE FIRST  INPUT ARGUMENT TO THCHSPP100
     2E CHSPPF SUBROUTINE IS OUTSIDE THE ALLOWABLE (0,1) INTERVAL *****)CHSPP101
   50 FORMAT (1H ,91H***** FATAL ERROR--THE SECOND INPUT ARGUMENT TO THECHSPP102
     2 CHSPPF SUBROUTINE IS NON-POSITIVE *****)                          CHSPP103
   60 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)CHSPP104
   70 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,I8,6H *****)   CHSPP105
C                                                                        CHSPP106
C-----START POINT-------------------------------------------------------CHSPP107
C                                                                        CHSPP108
C     EXPRESS THE CHI-SQUARED DISTRIBUTION PERCENT POINT                 CHSPP109
C     FUNCTION IN TERMS OF THE EQUIVALENT GAMMA                          CHSPP110
C     DISTRIBUTION PERCENT POINT FUNCTION,                               CHSPP111
C     AND THEN EVALUATE THE LATTER.                                      CHSPP112
C                                                                        CHSPP113
      ANU=NU                                                             CHSPP114
      GAMMA=ANU/2.0                                                      CHSPP115
```

```
      DP=P                                                              CHSPP116
      DNU=NU                                                            CHSPP117
      DGAMMA=DNU/2.0D0                                                  CHSPP118
      MAXIT=10000                                                       CHSPP119
C                                                                       CHSPP120
C     COMPUTE THE GAMMA FUNCTION USING THE ALGORITHM IN THE             CHSPP121
C     NBS APPLIED MATHEMATICS SERIES REFERENCE.                         CHSPP122
C     THIS GAMMA FUNCTION NEED BE CALCULATED ONLY ONCE.                 CHSPP123
C     IT IS USED IN THE CALCULATION OF THE CDF BASED ON                 CHSPP124
C     THE TENTATIVE VALUE OF THE PPF IN THE ITERATION.                  CHSPP125
C                                                                       CHSPP126
      Z=DGAMMA                                                          CHSPP127
      DEN=1.0D0                                                         CHSPP128
   80 IF (Z.GE.10.0D0) GO TO 90                                         CHSPP129
      DEN=DEN*Z                                                         CHSPP130
      Z=Z+1.0D0                                                         CHSPP131
      GO TO 80                                                          CHSPP132
   90 Z2=Z*Z                                                            CHSPP133
      Z3=Z*Z2                                                           CHSPP134
      Z4=Z2*Z2                                                          CHSPP135
      Z5=Z2*Z3                                                          CHSPP136
      A=(Z-0.5D0)*DLOG(Z)-Z+C                                           CHSPP137
      B=D(1)/Z+D(2)/Z3+D(3)/Z5+D(4)/(Z2*Z5)+D(5)/(Z4*Z5)+D(6)/(Z*Z5*Z5)+CHSPP138
     2D(7)/(Z3*Z5*Z5)+D(8)/(Z5*Z5*Z5)+D(9)/(Z2*Z5*Z5*Z5)                CHSPP139
      G=DEXP(A+B)/DEN                                                   CHSPP140
C                                                                       CHSPP141
C     DETERMINE LOWER AND UPPER LIMITS ON THE DESIRED 100P              CHSPP142
C     PERCENT POINT.                                                    CHSPP143
C                                                                       CHSPP144
      ILOOP=1                                                           CHSPP145
      XMIN0=(DP*DGAMMA*G)**(1.0D0/DGAMMA)                               CHSPP146
      XMIN=XMIN0                                                        CHSPP147
      ICOUNT=1                                                          CHSPP148
  100 AI=ICOUNT                                                         CHSPP149
      XMAX=AI*XMIN0                                                     CHSPP150
      DX=XMAX                                                           CHSPP151
      GO TO 180                                                         CHSPP152
  110 IF (PCALC.GE.DP) GO TO 120                                        CHSPP153
      XMIN=XMAX                                                         CHSPP154
      ICOUNT=ICOUNT+1                                                   CHSPP155
      IF (ICOUNT.LE.30000) GO TO 100                                    CHSPP156
  120 XMID=(XMIN+XMAX)/2.0D0                                            CHSPP157
C                                                                       CHSPP158
C     NOW ITERATE BY BISECTION UNTIL THE DESIRED ACCURACY IS ACHIEVED.  CHSPP159
C                                                                       CHSPP160
      ILOOP=2                                                           CHSPP161
      XLOWER=XMIN                                                       CHSPP162
      XUPPER=XMAX                                                       CHSPP163
      ICOUNT=0                                                          CHSPP164
  130 DX=XMID                                                           CHSPP165
      GO TO 180                                                         CHSPP166
  140 IF (PCALC.EQ.DP) GO TO 170                                        CHSPP167
      IF (PCALC.GT.DP) GO TO 150                                        CHSPP168
      XLOWER=XMID                                                       CHSPP169
      XMID=(XMID+XUPPER)/2.0D0                                          CHSPP170
      GO TO 160                                                         CHSPP171
  150 XUPPER=XMID                                                       CHSPP172
      XMID=(XMID+XLOWER)/2.0D0                                          CHSPP173
```

```
  160 XDEL=XMID-XLOWER                                                   CHSPP174
      IF (XDEL.LT.0.0D0) XDEL=-XDEL                                      CHSPP175
      ICOUNT=ICOUNT+1                                                    CHSPP176
      IF (XDEL.LT.0.0000000001D0.OR.ICOUNT.GT.100) GO TO 170             CHSPP177
      GO TO 130                                                          CHSPP178
  170 PPF=2.0D0*XMID                                                     CHSPP179
      RETURN                                                             CHSPP180
C                                                                        CHSPP181
C*********************************************************************** CHSPP182
C     THIS SECTION BELOW IS LOGICALLY SEPARATE FROM THE ABOVE.           CHSPP183
C     THIS SECTION COMPUTES A CDF VALUE FOR ANY GIVEN TENTATIVE          CHSPP184
C     PERCENT POINT X VALUE AS DEFINED IN EITHER OF THE 2                CHSPP185
C     ITERATION LOOPS IN THE ABOVE CODE.                                 CHSPP186
C                                                                        CHSPP187
C     COMPUTE T-SUB-Q AS DEFINED ON PAGE 4 OF THE WILK, GNANADESIKAN,    CHSPP188
C     AND HUYETT REFERENCE                                               CHSPP189
C                                                                        CHSPP190
  180 SUM=1.0D0/DGAMMA                                                   CHSPP191
      TERM=1.0D0/DGAMMA                                                  CHSPP192
      CUT1=DX-DGAMMA                                                     CHSPP193
      CUT2=DX*10000000000.0D0                                            CHSPP194
      DO 190 J=1,MAXIT                                                   CHSPP195
      AJ=J                                                               CHSPP196
      TERM=DX*TERM/(DGAMMA+AJ)                                           CHSPP197
      SUM=SUM+TERM                                                       CHSPP198
      CUTOFF=CUT1+(CUT2*TERM/SUM)                                        CHSPP199
      IF (AJ.GT.CUTOFF) GO TO 200                                        CHSPP200
  190 CONTINUE                                                           CHSPP201
      WRITE (IPR,210) MAXIT                                              CHSPP202
      WRITE (IPR,220) P                                                  CHSPP203
      WRITE (IPR,230) NU                                                 CHSPP204
      WRITE (IPR,240)                                                    CHSPP205
      PPF=0.0                                                            CHSPP206
      RETURN                                                             CHSPP207
C                                                                        CHSPP208
  200 T=SUM                                                              CHSPP209
      PCALC=(DX**DGAMMA)*(DEXP(-DX))*T/G                                 CHSPP210
      IF (ILOOP.EQ.1) GO TO 110                                          CHSPP211
      GO TO 140                                                          CHSPP212
C                                                                        CHSPP213
  210 FORMAT (1H ,48H*****ERROR IN INTERNAL OPERATIONS IN THE CHSPPF ,   CHSPP214
     2 45HSUBROUTINE--THE NUMBER OF ITERATIONS EXCEEDS ,I7)              CHSPP215
  220 FORMAT (1H ,33H      THE INPUT VALUE OF P      IS ,E15.8)          CHSPP216
  230 FORMAT (1H ,33H      THE INPUT VALUE OF NU     IS ,I8)             CHSPP217
  240 FORMAT (1H ,48H      THE OUTPUT VALUE OF PPF HAS BEEN SET TO 0.0)  CHSPP218
C                                                                        CHSPP219
      END                                                                CHSPP220
```

```
CPR*NS(1).CIYFIN(6)
      SUBROUTINE CIYFIN (XF,YF,YFSD,NF,RSD,AL,DL,C,NRSD,NB,YFL,YFU,IP)  CIYFIN01
C-----------------------------------------------------------------------CIYFIN02
C     CIYFIN   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING     CIYFIN03
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. CIYFIN04
C              AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION CIYFIN05
C     FOR: COMPUTING CONFIDENCE INTERVALS FOR PREDICTED Y-VALUES ON A   CIYFIN06
C          CALIBRATION CURVE USING SCHEFFE'S TECHNIQUE                  CIYFIN07
C                                                                       CIYFIN08
C     REFERENCE: SCHEFFE, HENRY, 'A STATISTICAL THEORY OF CALIBRATION'  CIYFIN09
C                THE ANNALS OF STATISTICS, VOLUME 1, NUMBER 1           CIYFIN10
C                JANUARY 1973, PP. 1-37                                 CIYFIN11
C                                                                       CIYFIN12
C     SUBPROGRAMS CALLED: CHSPPF, FPPF, NORPPF                          CIYFIN13
C     CURRENT VERSION COMPLETED MARCH 17, 1980                          CIYFIN14
C-----------------------------------------------------------------------CIYFIN15
      DIMENSION XF(NF),YF(NF),YFSD(NF),YFL(NF),YFU(NF)                  CIYFIN16
C--- WRITE FORMATS                                                      CIYFIN17
10    FORMAT (15X,2HZ(,F7.5,3H) =,F11.5)                                CIYFIN18
20    FORMAT (6X,6HCHISQ(,F7.5,1H,,I4,3H) =,F11.5)                      CIYFIN19
30    FORMAT (5X,2HF(,F7.5,1H,,I4,1H,,I4,3H) =,F11.5)                   CIYFIN20
40    FORMAT (//1X,75(1H-)/1X,26H* CONFIDENCE INTERVALS FOR,I5,1X,      CIYFIN21
     2 43HEVENLY SPACED POINTS WITHIN THE KNOT SPAN */1X,75(1H-)//5X,   CIYFIN22
     3 7HALPHA =,F8.5,5X,7HDELTA =,F8.5,5X,3HC =,F5.2/)                 CIYFIN23
50    FORMAT (1X,I4,5G13.6)                                             CIYFIN24
60    FORMAT (//20X,9HPREDICTED,5X,7HSTD DEV,6X,19HCONFIDENCE INTERVAL/ CIYFIN25
     2 4X,1HI,5X,4HX(I),9X,4HY(I),6X,9HPRED Y(I),6X,5HLOWER,8X,5HUPPER/)CIYFIN26
      WRITE (6,40) NF,AL,DL,C                                           CIYFIN27
C--- COMPUTE Z(1-AL/2) CRITICAL POINT FOR N(0,1) P.D.F.                 CIYFIN28
      P=1.0-AL/2.0                                                      CIYFIN29
      CALL NORPPF (P,ZAL)                                               CIYFIN30
      WRITE (6,10) P,ZAL                                                CIYFIN31
C--- ARTIFICIALLY SET NEXT TWO CRITICAL POINTS IF DELTA=1               CIYFIN32
      CDL=NRSD                                                          CIYFIN33
      FDL=0                                                             CIYFIN34
      P=1.0-DL                                                          CIYFIN35
      IF (DL.EQ.1.0) GO TO 70                                           CIYFIN36
C--- COMPUTE CHISQ(DL) CRITICAL POINT FOR CHI-SQUARED(NRSD) P.D.F.      CIYFIN37
      CALL CHSPPF (DL,NRSD,CDL)                                         CIYFIN38
      WRITE (6,20) DL,NRSD,CDL                                          CIYFIN39
C--- COMPUTE F(1-DL) CRITICAL POINT FOR F(NB,NRSD) P.D.F.               CIYFIN40
      CALL FPPF (P,NB,NRSD,FDL)                                         CIYFIN41
70    WRITE (6,30) P,NB,NRSD,FDL                                        CIYFIN42
C--- COMPUTE CONFIDENCE INTERVAL FOR EACH Y VALUE                       CIYFIN43
      C1=ZAL*SQRT(FLOAT(NRSD)/CDL)                                      CIYFIN44
      C2=SQRT(FLOAT(NB)*FDL)                                            CIYFIN45
      C3=C*RSD                                                          CIYFIN46
      DO 80 I=1,NF                                                      CIYFIN47
      WIDTH=C3*(C1+C2*YFSD(I))                                          CIYFIN48
      YFL(I)=YF(I)-WIDTH                                                CIYFIN49
      YFU(I)=YF(I)+WIDTH                                                CIYFIN50
80    CONTINUE                                                          CIYFIN51
C--- CHECK WHETHER TO PRINT OUT EACH Y VALUE AND ITS STANDARD DEVIATION CIYFIN52
      IF (IP.LT.2) GO TO 100                                            CIYFIN53
C--- PRINT OUT EACH Y VALUE AND ITS STANDARD DEVIATION                  CIYFIN54
      WRITE (6,60)                                                      CIYFIN55
      DO 90 I=1,NF                                                      CIYFIN56
      WRITE (6,50) I,XF(I),YF(I),YFSD(I),YFL(I),YFU(I)                  CIYFIN57
```

```
   90   CONTINUE                                                         CIYFIN58
        RETURN                                                           CIYFIN59
  100   WRITE (6,110)                                                    CIYFIN60
  110   FORMAT (/1X,43H******* PRINTOUT OF Y CONFIDENCE INTERVALS ,      CIYFIN61
       2 18HSUPPRESSED *******)                                          CIYFIN62
        RETURN                                                           CIYFIN63
        END                                                              CIYFIN64
```

```
CPR*NS(1).COVAR(2)
      SUBROUTINE COVAR (NMX,N,KMX,K,Q,CI)                               COVAR001
C                                                                       COVAR002
      INTEGER NMX,N,KMX,K                                               COVAR003
      REAL Q(KMX,N),CI(NMX,N)                                           COVAR004
C                                                                       COVAR005
C  THIS FORTRAN SUBROUTINE COMPUTES AND RETURNS THE N X N UNSCALED      COVAR006
C  COVARIANCE MATRIX CI OBTAINED BY INVERTING THE GRAMIAN MATRIX C. THE COVAR007
C  CHOLESKY FACTOR L OF C IS ASSUMED TO BE STORED IN Q ON INPUT.        COVAR008
C  SUBROUTINE BCHSLV IS USED TO SOLVE FOR EACH COLUMN OF THE INVERSE.   COVAR009
C                                                                       COVAR010
C  ON INPUT.                                                            COVAR011
C                                                                       COVAR012
C     NMX            IS THE ROW DIMENSION OF CI.                        COVAR013
C                                                                       COVAR014
C     N              IS THE DIMENSION OF THE SPACE OF SPLINES OF ORDER  COVAR015
C                    K.                                                 COVAR016
C                                                                       COVAR017
C     KMX            IS THE ROW DIMENSION OF Q.                         COVAR018
C                                                                       COVAR019
C     K              IS THE ORDER OF THE SPLINES = DEGREE + 1           COVAR020
C                                                                       COVAR021
C     Q(*, *)        HAS ROW DIMENSION KMX AND COLUMN DIMENSION AT LEASTCOVAR022
C                    N. THE CHOLESKY FACTOR L OF C IS STORED IN THE     COVAR023
C                    FIRST K ROWS OF THE MATRIX.                        COVAR024
C                                                                       COVAR025
C  ON OUTPUT.                                                           COVAR026
C                                                                       COVAR027
C     CI(*, *)       HAS ROW DIMENSION NMX AND COLUMN DIMENSION AT LEASTCOVAR028
C                    N. IT CONTAINS THE UNSCALED COVARIANCE MATRIX IN   COVAR029
C                    STANDARD ROW, COLUMN FORM.                         COVAR030
C                                                                       COVAR031
C     AND THE REST OF THE VARIABLES ARE UNCHANGED.                      COVAR032
C                                                                       COVAR033
C  ADDITIONAL ROUTINES REQUIRED.                                        COVAR034
C                                                                       COVAR035
C     BCHSLV                                                            COVAR036
C                                                                       COVAR037
C  BY.                                                                  COVAR038
C                                                                       COVAR039
C     MARTIN CORDES                                                     COVAR040
C     CENTER FOR APPLIED MATHEMATICS, NBS                               COVAR041
C     VERSION 1 - OCT 1979                                              COVAR042
C                                                                       COVAR043
C-----------------------------------------------------------------------COVAR044
C                                                                       COVAR045
      INTEGER I,J                                                       COVAR046
C                                                                       COVAR047
      DO 20 J=1,N                                                       COVAR048
      DO 10 I=1,N                                                       COVAR049
      CI(I,J)=0.0                                                       COVAR050
   10 CONTINUE                                                          COVAR051
      CI(J,J)=1.0                                                       COVAR052
      CALL BCHSLV (Q,KMX,K,N,CI(1,J))                                   COVAR053
   20 CONTINUE                                                          COVAR054
C                                                                       COVAR055
      RETURN                                                            COVAR056
      END                                                               COVAR057
```

CPR*NS(1).FCDF(1)

```
      SUBROUTINE FCDF (X,NU1,NU2,CDF)                                   FCDF0001
C                                                                       FCDF0002
C     PURPOSE--THIS SUBROUTINE COMPUTES THE CUMULATIVE DISTRIBUTION     FCDF0003
C              FUNCTION VALUE FOR THE F DISTRIBUTION                    FCDF0004
C              WITH INTEGER DEGREES OF FREEDOM                          FCDF0005
C              PARAMETERS = NU1 AND NU2.                                FCDF0006
C              THIS DISTRIBUTION IS DEFINED FOR ALL NON-NEGATIVE X.     FCDF0007
C              THE PROBABILITY DENSITY FUNCTION IS GIVEN                FCDF0008
C              IN THE REFERENCES BELOW.                                 FCDF0009
C     INPUT  ARGUMENTS--X      = THE SINGLE PRECISION VALUE AT          FCDF0010
C                                WHICH THE CUMULATIVE DISTRIBUTION      FCDF0011
C                                FUNCTION IS TO BE EVALUATED.           FCDF0012
C                                X SHOULD BE NON-NEGATIVE.              FCDF0013
C                     --NU1    = THE INTEGER DEGREES OF FREEDOM         FCDF0014
C                                FOR THE NUMERATOR OF THE F RATIO.      FCDF0015
C                                NU1 SHOULD BE POSITIVE.                FCDF0016
C                     --NU2    = THE INTEGER DEGREES OF FREEDOM         FCDF0017
C                                FOR THE DENOMINATOR OF THE F RATIO.    FCDF0018
C                                NU2 SHOULD BE POSITIVE.                FCDF0019
C     OUTPUT ARGUMENTS--CDF    = THE SINGLE PRECISION CUMULATIVE        FCDF0020
C                                DISTRIBUTION FUNCTION VALUE.           FCDF0021
C     OUTPUT--THE SINGLE PRECISION CUMULATIVE DISTRIBUTION              FCDF0022
C             FUNCTION VALUE CDF FOR THE F DISTRIBUTION                 FCDF0023
C             WITH DEGREES OF FREEDOM                                   FCDF0024
C             PARAMETERS = NU1 AND NU2.                                 FCDF0025
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.   FCDF0026
C     RESTRICTIONS--X SHOULD BE NON-NEGATIVE.                           FCDF0027
C                 --NU1 SHOULD BE A POSITIVE INTEGER VARIABLE.          FCDF0028
C                 --NU2 SHOULD BE A POSITIVE INTEGER VARIABLE.          FCDF0029
C     OTHER DATAPAC   SUBROUTINES NEEDED--NORCDF,CHSCDF.                FCDF0030
C     FORTRAN LIBRARY SUBROUTINES NEEDED--DSQRT, DATAN.                 FCDF0031
C     MODE OF INTERNAL OPERATIONS--DOUBLE PRECISION.                    FCDF0032
C     LANGUAGE--ANSI FORTRAN.                                           FCDF0033
C     REFERENCES--NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS      FCDF0034
C                 SERIES 55, 1964, PAGES 946-947,                       FCDF0035
C                 FORMULAE 26.6.4, 26.6.5, 26.6.8, AND 26.6.15.         FCDF0036
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE               FCDF0037
C                 DISTRIBUTIONS--2, 1970, PAGE 83, FORMULA 20,          FCDF0038
C                 AND PAGE 84, THIRD FORMULA.                           FCDF0039
C               --PAULSON, AN APPROXIMATE NORMAILIZATION                FCDF0040
C                 OF THE ANALYSIS OF VARIANCE DISTRIBUTION,             FCDF0041
C                 ANNALS OF MATHEMATICAL STATISTICS, 1942,              FCDF0042
C                 NUMBER 13, PAGES 233-135.                             FCDF0043
C               --SCHEFFE AND TUKEY, A FORMULA FOR SAMPLE SIZES         FCDF0044
C                 FOR POPULATION TOLERANCE LIMITS, 1944,                FCDF0045
C                 NUMBER 15, PAGE 217.                                  FCDF0046
C     WRITTEN BY--JAMES J. FILLIBEN                                     FCDF0047
C                 STATISTICAL ENGINEERING LABORATORY (205.03)           FCDF0048
C                 NATIONAL BUREAU OF STANDARDS                          FCDF0049
C                 WASHINGTON, D. C. 20234                               FCDF0050
C                 PHONE:  301-921-2315                                  FCDF0051
C     ORIGINAL VERSION--AUGUST    1972.                                 FCDF0052
C     UPDATED         --SEPTEMBER 1975.                                 FCDF0053
C     UPDATED         --NOVEMBER  1975.                                 FCDF0054
C     UPDATED         --OCTOBER   1976.                                 FCDF0055
C                                                                       FCDF0056
C-----------------------------------------------------------------------FCDF0057
```

```
C                                                                            FCDF0058
      DOUBLE PRECISION DX,PI,ANU1,ANU2,Z,SUM,TERM,AI,COEF1,COEF2,ARG         FCDF0059
      DOUBLE PRECISION COEF                                                  FCDF0060
      DOUBLE PRECISION THETA,SINTH,COSTH,A,B                                 FCDF0061
      DOUBLE PRECISION DSQRT,DATAN                                           FCDF0062
      DOUBLE PRECISION DFACT1,DFACT2,DNUM,DDEN                               FCDF0063
      DOUBLE PRECISION DPOW1,DPOW2                                           FCDF0064
      DOUBLE PRECISION DNU1,DNU2                                             FCDF0065
      DOUBLE PRECISION TERM1,TERM2,TERM3                                     FCDF0066
      DATA PI /3.14159265358979D0/                                           FCDF0067
      DATA DPOW1,DPOW2 /0.33333333333333D0,0.66666666666667D0/               FCDF0068
      DATA NUCUT1,NUCUT2 /100,1000/                                          FCDF0069
C                                                                            FCDF0070
      IPR=6                                                                  FCDF0071
C                                                                            FCDF0072
C     CHECK THE INPUT ARGUMENTS FOR ERRORS                                   FCDF0073
C                                                                            FCDF0074
      IF (NU1.LE.0) GO TO 10                                                 FCDF0075
      IF (NU2.LE.0) GO TO 20                                                 FCDF0076
      IF (X.LT.0.0) GO TO 30                                                 FCDF0077
      GO TO 40                                                               FCDF0078
   10 WRITE (IPR,60)                                                         FCDF0079
      WRITE (IPR,90) NU1                                                     FCDF0080
      CDF=0.0                                                                FCDF0081
      RETURN                                                                 FCDF0082
   20 WRITE (IPR,70)                                                         FCDF0083
      WRITE (IPR,90) NU2                                                     FCDF0084
      CDF=0.0                                                                FCDF0085
      RETURN                                                                 FCDF0086
   30 WRITE (IPR,50)                                                         FCDF0087
      WRITE (IPR,80) X                                                       FCDF0088
      CDF=0.0                                                                FCDF0089
      RETURN                                                                 FCDF0090
   40 CONTINUE                                                               FCDF0091
   50 FORMAT (1H ,96H***** NON-FATAL DIAGNOSTIC--THE FIRST  INPUT ARGUMEFCDF0092
     2NT TO THE FCDF   SUBROUTINE IS NEGATIVE *****)                         FCDF0093
   60 FORMAT (1H ,91H***** FATAL ERROR--THE SECOND INPUT ARGUMENT TO THEFCDF0094
     2 FCDF   SUBROUTINE IS NON-POSITIVE *****)                              FCDF0095
   70 FORMAT (1H ,91H***** FATAL ERROR--THE THIRD  INPUT ARGUMENT TO THEFCDF0096
     2 FCDF   SUBROUTINE IS NON-POSITIVE *****)                              FCDF0097
   80 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)FCDF0098
   90 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,I8,6H *****)   FCDF0099
C                                                                            FCDF0100
C-----START POINT-----------------------------------------------------       FCDF0101
C                                                                            FCDF0102
      DX=X                                                                   FCDF0103
      M=NU1                                                                  FCDF0104
      N=NU2                                                                  FCDF0105
      ANU1=NU1                                                               FCDF0106
      ANU2=NU2                                                               FCDF0107
      DNU1=NU1                                                               FCDF0108
      DNU2=NU2                                                               FCDF0109
C                                                                            FCDF0110
C     IF X IS NON-POSITIVE, SET CDF = 0.0 AND RETURN.                        FCDF0111
C     IF NU2 IS 5 THROUGH 9 AND X IS MORE THAN 3000                          FCDF0112
C     STANDARD DEVIATIONS BELOW THE MEAN,                                    FCDF0113
C     SET CDF = 0.0 AND RETURN.                                              FCDF0114
C     IF NU2 IS 10 OR LARGER AND X IS MORE THAN 150                          FCDF0115
```

```
C     STANDARD DEVIATIONS BELOW THE MEAN,                               FCDF0116
C     SET CDF = 0.0 AND RETURN.                                         FCDF0117
C     IF NU2 IS 5 THROUGH 9 AND X IS MORE THAN 3000                     FCDF0118
C     STANDARD DEVIATIONS ABOVE THE MEAN,                               FCDF0119
C     SET CDF = 1.0 AND RETURN.                                         FCDF0120
C     IF NU2 IS 10 OR LARGER AND X IS MORE THAN 150                     FCDF0121
C     STANDARD DEVIATIONS ABOVE THE MEAN,                               FCDF0122
C     SET CDF = 1.0 AND RETURN.                                         FCDF0123
C                                                                       FCDF0124
      IF (X.LE.0.0) GO TO 100                                           FCDF0125
      IF (NU2.LE.4) GO TO 120                                           FCDF0126
      T1=2.0/ANU1                                                       FCDF0127
      T2=ANU2/(ANU2-2.0)                                                FCDF0128
      T3=(ANU1+ANU2-2.0)/(ANU2-4.0)                                     FCDF0129
      AMEAN=T2                                                          FCDF0130
      SD=SQRT(T1*T2*T2*T3)                                              FCDF0131
      ZRATIO=(X-AMEAN)/SD                                               FCDF0132
      IF (NU2.LT.10.AND.ZRATIO.LT.-3000.0) GO TO 100                    FCDF0133
      IF (NU2.GE.10.AND.ZRATIO.LT.-150.0) GO TO 100                     FCDF0134
      IF (NU2.LT.10.AND.ZRATIO.GT.3000.0) GO TO 110                     FCDF0135
      IF (NU2.GE.10.AND.ZRATIO.GT.150.0) GO TO 110                      FCDF0136
      GO TO 120                                                         FCDF0137
  100 CDF=0.0                                                           FCDF0138
      RETURN                                                            FCDF0139
  110 CDF=1.0                                                           FCDF0140
      RETURN                                                            FCDF0141
  120 CONTINUE                                                          FCDF0142
C                                                                       FCDF0143
C     DISTINGUISH BETWEEN 6 SEPARATE REGIONS                            FCDF0144
C     OF THE (NU1,NU2) SPACE.                                           FCDF0145
C     BRANCH TO THE PROPER COMPUTATIONAL METHOD                         FCDF0146
C     DEPENDING ON THE REGION.                                          FCDF0147
C     NUCUT1 HAS THE VALUE 100.                                         FCDF0148
C     NUCUT2 HAS THE VALUE 1000.                                        FCDF0149
C                                                                       FCDF0150
      IF (NU1.LT.NUCUT2.AND.NU2.LT.NUCUT2) GO TO 140                    FCDF0151
      IF (NU1.GE.NUCUT2.AND.NU2.GE.NUCUT2) GO TO 310                    FCDF0152
      IF (NU1.LT.NUCUT1.AND.NU2.GE.NUCUT2) GO TO 320                    FCDF0153
      IF (NU1.GE.NUCUT1.AND.NU2.GE.NUCUT2) GO TO 310                    FCDF0154
      IF (NU1.GE.NUCUT2.AND.NU2.LT.NUCUT1) GO TO 330                    FCDF0155
      IF (NU1.GE.NUCUT2.AND.NU2.GE.NUCUT1) GO TO 310                    FCDF0156
      IBRAN=5                                                           FCDF0157
      WRITE (IPR,130) IBRAN                                             FCDF0158
  130 FORMAT (1H ,42H*****INTERNAL ERROR IN   FCDF SUBROUTINE--,        FCDF0159
     2 46HIMPOSSIBLE BRANCH CONDITION AT BRANCH POINT = ,I8)            FCDF0160
      RETURN                                                            FCDF0161
C                                                                       FCDF0162
C     TREAT THE CASE WHEN NU1 AND NU2                                   FCDF0163
C     ARE BOTH SMALL OR MODERATE                                        FCDF0164
C     (THAT IS, BOTH ARE SMALLER THAN 1000).                            FCDF0165
C     METHOD UTILIZED--EXACT FINITE SUM                                 FCDF0166
C     (SEE AMS 55, PAGE 946, FORMULAE 26.6.4, 26.6.5,                   FCDF0167
C     AND 26.6.8).                                                      FCDF0168
C                                                                       FCDF0169
  140 CONTINUE                                                          FCDF0170
      Z=ANU2/(ANU2+ANU1*DX)                                             FCDF0171
      IFLAG1=NU1-2*(NU1/2)                                              FCDF0172
      IFLAG2=NU2-2*(NU2/2)                                              FCDF0173
```

```
      IF (IFLAG1.EQ.0) GO TO 150                                        FCDF0174
      IF (IFLAG2.EQ.0) GO TO 180                                        FCDF0175
      GO TO 210                                                         FCDF0176
C                                                                       FCDF0177
C     DO THE NU1 EVEN AND NU2 EVEN OR ODD CASE                          FCDF0178
C                                                                       FCDF0179
  150 SUM=0.0D0                                                         FCDF0180
      TERM=1.0D0                                                        FCDF0181
      IMAX=(M-2)/2                                                      FCDF0182
      IF (IMAX.LE.0) GO TO 170                                          FCDF0183
      DO 160 I=1,IMAX                                                   FCDF0184
      AI=I                                                              FCDF0185
      COEF1=2.0D0*(AI-1.0D0)                                            FCDF0186
      COEF2=2.0D0*AI                                                    FCDF0187
      TERM=TERM*((ANU2+COEF1)/COEF2)*(1.0D0-Z)                          FCDF0188
      SUM=SUM+TERM                                                      FCDF0189
  160 CONTINUE                                                          FCDF0190
C                                                                       FCDF0191
  170 SUM=SUM+1.0D0                                                     FCDF0192
      SUM=(Z**(ANU2/2.0D0))*SUM                                         FCDF0193
      CDF=1.0D0-SUM                                                     FCDF0194
      RETURN                                                            FCDF0195
C                                                                       FCDF0196
C     DO THE NU1 ODD AND NU2 EVEN CASE                                  FCDF0197
C                                                                       FCDF0198
  180 SUM=0.0D0                                                         FCDF0199
      TERM=1.0D0                                                        FCDF0200
      IMAX=(N-2)/2                                                      FCDF0201
      IF (IMAX.LE.0) GO TO 200                                          FCDF0202
      DO 190 I=1,IMAX                                                   FCDF0203
      AI=I                                                              FCDF0204
      COEF1=2.0D0*(AI-1.0D0)                                            FCDF0205
      COEF2=2.0D0*AI                                                    FCDF0206
      TERM=TERM*((ANU1+COEF1)/COEF2)*Z                                  FCDF0207
      SUM=SUM+TERM                                                      FCDF0208
  190 CONTINUE                                                          FCDF0209
C                                                                       FCDF0210
  200 SUM=SUM+1.0D0                                                     FCDF0211
      CDF=((1.0D0-Z)**(ANU1/2.0D0))*SUM                                 FCDF0212
      RETURN                                                            FCDF0213
C                                                                       FCDF0214
C     DO THE NU1 ODD AND NU2 ODD CASE                                   FCDF0215
C                                                                       FCDF0216
  210 SUM=0.0D0                                                         FCDF0217
      TERM=1.0D0                                                        FCDF0218
      ARG=DSQRT((ANU1/ANU2)*DX)                                         FCDF0219
      THETA=DATAN(ARG)                                                  FCDF0220
      SINTH=ARG/DSQRT(1.0D0+ARG*ARG)                                    FCDF0221
      COSTH=1.0D0/DSQRT(1.0D0+ARG*ARG)                                  FCDF0222
      IF (N.EQ.1) GO TO 240                                             FCDF0223
      IF (N.EQ.3) GO TO 230                                             FCDF0224
      IMAX=N-2                                                          FCDF0225
      DO 220 I=3,IMAX,2                                                 FCDF0226
      AI=I                                                              FCDF0227
      COEF1=AI-1.0D0                                                    FCDF0228
      COEF2=AI                                                          FCDF0229
      TERM=TERM*(COEF1/COEF2)*(COSTH*COSTH)                             FCDF0230
      SUM=SUM+TERM                                                      FCDF0231
```

```
220   CONTINUE                                                 FCDF0232
C                                                              FCDF0233
230   SUM=SUM+1.0D0                                            FCDF0234
      SUM=SUM*SINTH*COSTH                                      FCDF0235
C                                                              FCDF0236
240   A=(2.0D0/PI)*(THETA+SUM)                                 FCDF0237
      SUM=0.0D0                                                FCDF0238
      TERM=1.0D0                                               FCDF0239
      IF (M.EQ.1) B=0.0D0                                      FCDF0240
      IF (M.EQ.1) GO TO 300                                    FCDF0241
      IF (M.EQ.3) GO TO 260                                    FCDF0242
      IMAX=M-3                                                 FCDF0243
      DO 250 I=1,IMAX,2                                        FCDF0244
      AI=I                                                     FCDF0245
      COEF1=AI                                                 FCDF0246
      COEF2=AI+2.0D0                                           FCDF0247
      TERM=TERM*((ANU2+COEF1)/COEF2)*(SINTH*SINTH)             FCDF0248
      SUM=SUM+TERM                                             FCDF0249
250   CONTINUE                                                 FCDF0250
C                                                              FCDF0251
260   SUM=SUM+1.0D0                                            FCDF0252
      SUM=SUM*SINTH*(COSTH**N)                                 FCDF0253
      COEF=1.0D0                                               FCDF0254
      IEVODD=N-2*(N/2)                                         FCDF0255
      IMIN=3                                                   FCDF0256
      IF (IEVODD.EQ.0) IMIN=2                                  FCDF0257
      IF (IMIN.GT.N) GO TO 280                                 FCDF0258
      DO 270 I=IMIN,N,2                                        FCDF0259
      AI=I                                                     FCDF0260
      COEF=((AI-1.0D0)/AI)*COEF                                FCDF0261
270   CONTINUE                                                 FCDF0262
C                                                              FCDF0263
280   COEF=COEF*ANU2                                           FCDF0264
      IF (IEVODD.EQ.0) GO TO 290                               FCDF0265
      COEF=COEF*(2.0D0/PI)                                     FCDF0266
C                                                              FCDF0267
290   B=COEF*SUM                                               FCDF0268
C                                                              FCDF0269
300   CDF=A-B                                                  FCDF0270
      RETURN                                                   FCDF0271
C                                                              FCDF0272
C     TREAT THE CASE WHEN NU1 AND NU2                          FCDF0273
C     ARE BOTH LARGE                                           FCDF0274
C     (THAT IS, BOTH ARE EQUAL TO OR LARGER THAN 1000);        FCDF0275
C     OR WHEN NU1 IS MODERATE AND NU2 IS LARGE                 FCDF0276
C     (THAT IS, WHEN NU1 IS EQUAL TO OR GREATER THAN 100       FCDF0277
C     BUT SMALLER THAN 1000,                                   FCDF0278
C     AND NU2 IS EQUAL TO OR LARGER THAN 1000);                FCDF0279
C     OR WHEN NU2 IS MODERATE AND NU1 IS LARGE                 FCDF0280
C     (THAT IS WHEN NU2 IS EQUAL TO OR GREATER THAN 100        FCDF0281
C     BUT SMALLER THAN 1000,                                   FCDF0282
C     AND NU1 IS EQUAL TO OR LARGER THAN 1000).                FCDF0283
C     METHOD UTILIZED--PAULSON APPROXIMATION                   FCDF0284
C     (SEE AMS 55, PAGE 947, FORMULA 26.6.15).                 FCDF0285
C                                                              FCDF0286
310   CONTINUE                                                 FCDF0287
      DFACT1=1.0D0/(4.5D0*DNU1)                                FCDF0288
      DFACT2=1.0D0/(4.5D0*DNU2)                                FCDF0289
```

```
      DNUM=((1.0D0-DFACT2)*(DX**DPOW1))-(1.0D0-DFACT1)                 FCDF0290
      DDEN=DSQRT((DFACT2*(DX**DPOW2))+DFACT1)                          FCDF0291
      U=DNUM/DDEN                                                      FCDF0292
      CALL NORCDF (U,GCDF)                                             FCDF0293
      CDF=GCDF                                                         FCDF0294
      RETURN                                                           FCDF0295
C                                                                      FCDF0296
C     TREAT THE CASE WHEN NU1 IS SMALL                                 FCDF0297
C     AND NU2 IS LARGE                                                 FCDF0298
C     (THAT IS, WHEN NU1 IS SMALLER THAN 100,                          FCDF0299
C     AND NU2 IS EQUAL TO OR LARGER THAN 1000).                        FCDF0300
C     METHOD UTILIZED--SHEFFE-TUKEY APPROXIMATION                      FCDF0301
C     (SEE JOHNSON AND KOTZ, VOLUME 2, PAGE 84, THIRD FORMULA).        FCDF0302
C                                                                      FCDF0303
  320 CONTINUE                                                         FCDF0304
      TERM1=DNU1                                                       FCDF0305
      TERM2=(DNU1/DNU2)*(0.5D0*DNU1-1.0D0)                             FCDF0306
      TERM3=-(DNU1/DNU2)*0.5D0                                         FCDF0307
      U=(TERM1+TERM2)/((1.0D0/DX)-TERM3)                               FCDF0308
      CALL CHSCDF (U,NU1,CCDF)                                         FCDF0309
      CDF=CCDF                                                         FCDF0310
      RETURN                                                           FCDF0311
C                                                                      FCDF0312
C     TREAT THE CASE WHEN NU2 IS SMALL                                 FCDF0313
C     AND NU1 IS LARGE                                                 FCDF0314
C     (THAT IS, WHEN NU2 IS SMALLER THAN 100,                          FCDF0315
C     AND NU1 IS EQUAL TO OR LARGER THAN 1000).                        FCDF0316
C     METHOD UTILIZED--SHEFFE-TUKEY APPROXIMATION                      FCDF0317
C     (SEE JOHNSON AND KOTZ, VOLUME 2, PAGE 84, THIRD FORMULA).        FCDF0318
C                                                                      FCDF0319
  330 CONTINUE                                                         FCDF0320
      TERM1=DNU2                                                       FCDF0321
      TERM2=(DNU2/DNU1)*(0.5D0*DNU2-1.0D0)                             FCDF0322
      TERM3=-(DNU2/DNU1)*0.5D0                                         FCDF0323
      U=(TERM1+TERM2)/(DX-TERM3)                                       FCDF0324
      CALL CHSCDF (U,NU2,CCDF)                                         FCDF0325
      CDF=1.0-CCDF                                                     FCDF0326
      RETURN                                                           FCDF0327
C                                                                      FCDF0328
      END                                                              FCDF0329
```

CPR*NS(1).FPPF(1)

```
      SUBROUTINE FPPF (P,NU1,NU2,PPF)                                   FPPF0001
C                                                                       FPPF0002
C     PURPOSE--THIS SUBROUTINE COMPUTES THE PERCENT POINT               FPPF0003
C              FOR THE F DISTRIBUTION                                   FPPF0004
C              WITH INTEGER DEGREES OF FREEDOM                          FPPF0005
C              PARAMETERS = NU1 AND NU2.                                FPPF0006
C              THIS DISTRIBUTION IS DEFINED FOR ALL NON-NEGATIVE X.     FPPF0007
C              THE PROBABILITY DENSITY FUNCTION IS GIVEN                FPPF0008
C              IN THE REFERENCES BELOW.                                 FPPF0009
C     INPUT  ARGUMENTS--P      = THE SINGLE PRECISION VALUE             FPPF0010
C                                (BETWEEN 0.0 AND 1.0)                  FPPF0011
C                                AT WHICH THE PERCENT POINT             FPPF0012
C                                FUNCTION IS TO BE EVALUATED.           FPPF0013
C                     --NU1    = THE INTEGER DEGREES OF FREEDOM         FPPF0014
C                                FOR THE NUMERATOR OF THE F RATIO.      FPPF0015
C                                NU1 SHOULD BE POSITIVE.                FPPF0016
C                     --NU2    = THE INTEGER DEGREES OF FREEDOM         FPPF0017
C                                FOR THE DENOMINATOR OF THE F RATIO.    FPPF0018
C                                NU2 SHOULD BE POSITIVE.                FPPF0019
C     OUTPUT ARGUMENTS--PPF    = THE SINGLE PRECISION PERCENT POINT     FPPF0020
C                                FUNCTION VALUE.                        FPPF0021
C     OUTPUT--THE SINGLE PRECISION PERCENT POINT                        FPPF0022
C             FUNCTION VALUE PPF FOR THE F DISTRIBUTION                 FPPF0023
C             WITH DEGREES OF FREEDOM                                   FPPF0024
C             PARAMETERS = NU1 AND NU2.                                 FPPF0025
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.   FPPF0026
C     RESTRICTIONS--P SHOULD BE BETWEEN                                 FPPF0027
C                   0.0 (INCLUSIVELY) AND 1.0 (EXCLUSIVELY).            FPPF0028
C                 --NU1 SHOULD BE A POSITIVE INTEGER VARIABLE.          FPPF0029
C                 --NU2 SHOULD BE A POSITIVE INTEGER VARIABLE.          FPPF0030
C     OTHER DATAPAC   SUBROUTINES NEEDED--FCDF, NORCDF, CHSCDF, NORPPF. FPPF0031
C     FORTRAN LIBRARY SUBROUTINES NEEDED--DSQRT, DATAN.                 FPPF0032
C     MODE OF INTERNAL OPERATIONS--DOUBLE PRECISION.                    FPPF0033
C     LANGUAGE--ANSI FORTRAN.                                           FPPF0034
C     REFERENCES--NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS      FPPF0035
C                 SERIES 55, 1964, PAGES 946-947,                       FPPF0036
C                 FORMULAE 26.6.4, 26.6.5, 26.6.8, AND 26.6.15.         FPPF0037
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE               FPPF0038
C                 DISTRIBUTIONS--2, 1970, PAGE 83, FORMULA 20,          FPPF0039
C                 AND PAGE 84, THIRD FORMULA.                           FPPF0040
C               --PAULSON, AN APPROXIMATE NORMAILIZATION                FPPF0041
C                 OF THE ANALYSIS OF VARIANCE DISTRIBUTION,             FPPF0042
C                 ANNALS OF MATHEMATICAL STATISTICS, 1942,              FPPF0043
C                 NUMBER 13, PAGES 233-135.                             FPPF0044
C               --SCHEFFE AND TUKEY, A FORMULA FOR SAMPLE SIZES         FPPF0045
C                 FOR POPULATION TOLERANCE LIMITS, 1944,                FPPF0046
C                 NUMBER 15, PAGE 217.                                  FPPF0047
C     WRITTEN BY--JAMES J. FILLIBEN                                     FPPF0048
C                 STATISTICAL ENGINEERING LABORATORY (205.03)           FPPF0049
C                 NATIONAL BUREAU OF STANDARDS                          FPPF0050
C                 WASHINGTON, D. C. 20234                               FPPF0051
C                 PHONE:  301-921-2315                                  FPPF0052
C     ORIGINAL VERSION--MAY       1978.                                 FPPF0053
C     UPDATED         --AUGUST    1979.                                 FPPF0054
C                                                                       FPPF0055
C-----------------------------------------------------------------------FPPF0056
C                                                                       FPPF0057
```

```
      IPR=6                                                             FPPF0058
C                                                                       FPPF0059
C     CHECK THE INPUT ARGUMENTS FOR ERRORS                              FPPF0060
C                                                                       FPPF0061
      PPF=0.0                                                           FPPF0062
      IF (NU1.LE.0) GO TO 10                                            FPPF0063
      IF (NU2.LE.0) GO TO 20                                            FPPF0064
      IF (P.LT.0.0.OR.P.GE.1.0) GO TO 30                                FPPF0065
      GO TO 40                                                          FPPF0066
   10 WRITE (IPR,60)                                                    FPPF0067
      WRITE (IPR,90) NU1                                                FPPF0068
      PPF=0.0                                                           FPPF0069
      RETURN                                                            FPPF0070
   20 WRITE (IPR,70)                                                    FPPF0071
      WRITE (IPR,90) NU2                                                FPPF0072
      PPF=0.0                                                           FPPF0073
      RETURN                                                            FPPF0074
   30 WRITE (IPR,50)                                                    FPPF0075
      WRITE (IPR,80) P                                                  FPPF0076
      PPF=0.0                                                           FPPF0077
      RETURN                                                            FPPF0078
   40 CONTINUE                                                          FPPF0079
   50 FORMAT (1H ,113H***** FATAL ERROR--THE FIRST  INPUT ARGUMENT TO THFPPF0080
     2E FPPF SUBROUTINE IS OUTSIDE THE ALLOWABLE (0,1) INTERVAL *****)  FPPF0081
   60 FORMAT (1H ,91H***** FATAL ERROR--THE SECOND INPUT ARGUMENT TO THEFPPF0082
     2 FPPF   SUBROUTINE IS NON-POSITIVE *****)                         FPPF0083
   70 FORMAT (1H ,91H***** FATAL ERROR--THE THIRD  INPUT ARGUMENT TO THEFPPF0084
     2 FCDF   SUBROUTINE IS NON-POSITIVE *****)                         FPPF0085
   80 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)FPPF0086
   90 FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,I8,6H *****)   FPPF0087
C                                                                       FPPF0088
C-----START POINT-----------------------------------------------------  FPPF0089
C                                                                       FPPF0090
      IBUG=0.0                                                          FPPF0091
C                                                                       FPPF0092
      TOL=0.000001                                                      FPPF0093
      MAXIT=100                                                         FPPF0094
      XMIN=0.0                                                          FPPF0095
      XMAX=10.0**30                                                     FPPF0096
      XLOW=XMIN                                                         FPPF0097
      XUP=XMAX                                                          FPPF0098
C                                                                       FPPF0099
      ANU1=NU1                                                          FPPF0100
      ANU2=NU2                                                          FPPF0101
C                                                                       FPPF0102
      EXPF=0.5*((1.0/ANU2)-(1.0/ANU1))                                  FPPF0103
      SDF=SQRT(0.5*((1.0/ANU2)+(1.0/ANU1)))                             FPPF0104
      CALL NORPPF (P,ZN)                                                FPPF0105
      XN=EXPF+ZN*SDF                                                    FPPF0106
      XMID=EXP(2.0*XN)                                                  FPPF0107
      IF (IBUG.EQ.1) WRITE (6,100) XMID                                 FPPF0108
  100 FORMAT (1H ,7HXMID = ,E15.7)                                      FPPF0109
C                                                                       FPPF0110
      IF (P.EQ.0.0) GO TO 110                                           FPPF0111
      GO TO 120                                                         FPPF0112
  110 CONTINUE                                                          FPPF0113
      PPF=XMIN                                                          FPPF0114
      RETURN                                                            FPPF0115
```

```
120   CONTINUE                                                  FPPF0116
C                                                               FPPF0117
      ICOUNT=0                                                  FPPF0118
C                                                               FPPF0119
130   CONTINUE                                                  FPPF0120
      X=XMID                                                    FPPF0121
      CALL FCDF (X,NU1,NU2,PCALC)                               FPPF0122
      IF (PCALC.EQ.P) GO TO 190                                 FPPF0123
      IF (PCALC.GT.P) GO TO 160                                 FPPF0124
C                                                               FPPF0125
140   CONTINUE                                                  FPPF0126
      XLOW=XMID                                                 FPPF0127
      X=XMID*2.0                                                FPPF0128
      IF (X.GE.XUP) GO TO 150                                   FPPF0129
      XMID=X                                                    FPPF0130
      IF (IBUG.EQ.1) WRITE (6,100) XMID                         FPPF0131
      CALL FCDF (X,NU1,NU2,PCALC)                               FPPF0132
      IF (PCALC.EQ.P) GO TO 190                                 FPPF0133
      IF (PCALC.LT.P) GO TO 140                                 FPPF0134
      XUP=X                                                     FPPF0135
150   CONTINUE                                                  FPPF0136
      XMID=(XLOW+XUP)/2.0                                       FPPF0137
      IF (IBUG.EQ.1) WRITE (6,100) XMID                         FPPF0138
      GO TO 180                                                 FPPF0139
C                                                               FPPF0140
160   CONTINUE                                                  FPPF0141
      XUP=XMID                                                  FPPF0142
      X=XMID/2.0                                                FPPF0143
      IF (X.LE.XLOW) GO TO 170                                  FPPF0144
      XMID=X                                                    FPPF0145
      IF (IBUG.EQ.1) WRITE (6,100) XMID                         FPPF0146
      CALL FCDF (X,NU1,NU2,PCALC)                               FPPF0147
      IF (PCALC.EQ.P) GO TO 190                                 FPPF0148
      IF (PCALC.GT.P) GO TO 160                                 FPPF0149
      XLOW=X                                                    FPPF0150
170   CONTINUE                                                  FPPF0151
      XMID=(XLOW+XUP)/2.0                                       FPPF0152
      IF (IBUG.EQ.1) WRITE (6,100) XMID                         FPPF0153
      GO TO 180                                                 FPPF0154
C                                                               FPPF0155
180   CONTINUE                                                  FPPF0156
      XDEL=ABS(XMID-XLOW)                                       FPPF0157
      ICOUNT=ICOUNT+1                                           FPPF0158
      IF (XDEL.LT.TOL.OR.ICOUNT.GT.MAXIT) GO TO 190             FPPF0159
      GO TO 130                                                 FPPF0160
C                                                               FPPF0161
190   CONTINUE                                                  FPPF0162
      PPF=XMID                                                  FPPF0163
C                                                               FPPF0164
      RETURN                                                    FPPF0165
      END                                                       FPPF0166
```

```
CPR*NS(1).GETX(2)
      SUBROUTINE GETX (XF,YF,NF,Y,L,M,X,I,KS,KL)                        GETX0001
C-----------------------------------------------------------------------GETX0002
C    GETX   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING         GETX0003
C           DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.    GETX0004
C           AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION    GETX0005
C    FOR: THE INVERSE INTERPOLATION OF A CALIBRATION CURVE OR ITS       GETX0006
C         UPPER OR LOWER CONFIDENCE LIMIT WHEREBY AN X-VALUE IS         GETX0007
C         COMPUTED FOR A GIVEN Y-VALUE                                  GETX0008
C    SUBPROGRAMS CALLED: -NONE-                                         GETX0009
C    CURRENT VERSION COMPLETED JUNE 18, 1980                            GETX0010
C-----------------------------------------------------------------------GETX0011
      DIMENSION XF(NF),YF(NF)                                           GETX0012
   10 IF (Y.LT.YF(L+1)) GO TO 20                                        GETX0013
      IF (L+1.EQ.NF) GO TO 30                                           GETX0014
      L=L+1                                                             GETX0015
      GO TO 10                                                          GETX0016
   20 IF (L.EQ.0) GO TO 40                                              GETX0017
      C=(Y-YF(L))/(YF(L+1)-YF(L))                                       GETX0018
      X=C*(XF(L+1)-XF(L))+XF(L)                                         GETX0019
      I=1                                                               GETX0020
      RETURN                                                            GETX0021
   30 X=XF(NF)                                                          GETX0022
      I=(5+M)/2                                                         GETX0023
      KL=KL+1                                                           GETX0024
      RETURN                                                            GETX0025
   40 X=XF(1)                                                           GETX0026
      I=(5-M)/2                                                         GETX0027
      KS=KS+1                                                           GETX0028
      RETURN                                                            GETX0029
      END                                                               GETX0030
```

```
CPR*NS(1).INTERV(1)
      SUBROUTINE INTERV (XT,LXT,X,LEFT,MFLAG)                           INTERV01
C  FROM  * A PRACTICAL GUIDE TO SPLINES *  BY C. DE BOOR                INTERV02
COMPUTES  LEFT = MAX( I , 1 .LE. I .LE. LXT  .AND.  XT(I) .LE. X )  .   INTERV03
C                                                                       INTERV04
C******  I N P U T  ******                                              INTERV05
C  XT.....A REAL SEQUENCE, OF LENGTH  LXT , ASSUMED TO BE NONDECREASING INTERV06
C  LXT.....NUMBER OF TERMS IN THE SEQUENCE  XT .                        INTERV07
C  X.....THE POINT WHOSE LOCATION WITH RESPECT TO THE SEQUENCE  XT  IS  INTERV08
C        TO BE DETERMINED.                                              INTERV09
C                                                                       INTERV10
C******  O U T P U T  ******                                            INTERV11
C  LEFT, MFLAG.....BOTH INTEGERS, WHOSE VALUE IS                        INTERV12
C                                                                       INTERV13
C   1     -1      IF                X .LT.  XT(1)                       INTERV14
C   I      0      IF   XT(I)  .LE. X .LT. XT(I+1)                       INTERV15
C  LXT     1      IF  XT(LXT) .LE. X                                    INTERV16
C                                                                       INTERV17
C        IN PARTICULAR,  MFLAG = 0 IS THE 'USUAL' CASE.  MFLAG .NE. 0   INTERV18
C        INDICATES THAT  X  LIES OUTSIDE THE HALFOPEN INTERVAL          INTERV19
C        XT(1) .LE. Y .LT. XT(LXT) . THE ASYMMETRIC TREATMENT OF THE    INTERV20
C        INTERVAL IS DUE TO THE DECISION TO MAKE ALL PP FUNCTIONS CONT- INTERV21
C        INUOUS FROM THE RIGHT.                                         INTERV22
C                                                                       INTERV23
C******  M E T H O D  ******                                            INTERV24
C  THE PROGRAM IS DESIGNED TO BE EFFICIENT IN THE COMMON SITUATION THAT INTERV25
C  IT IS CALLED REPEATEDLY, WITH  X  TAKEN FROM AN INCREASING OR DECREA-INTERV26
C  SING SEQUENCE. THIS WILL HAPPEN, E.G., WHEN A PP FUNCTION IS TO BE   INTERV27
C  GRAPHED. THE FIRST GUESS FOR  LEFT  IS THEREFORE TAKEN TO BE THE VAL-INTERV28
C  UE RETURNED AT THE PREVIOUS CALL AND STORED IN THE  L O C A L  VARIA-INTERV29
C  BLE  ILO . A FIRST CHECK ASCERTAINS THAT  ILO .LT. LXT (THIS IS NEC- INTERV30
C  ESSARY SINCE THE PRESENT CALL MAY HAVE NOTHING TO DO WITH THE PREVI- INTERV31
C  OUS CALL). THEN, IF  XT(ILO) .LE. X .LT. XT(ILO+1), WE SET  LEFT =   INTERV32
C  ILO  AND ARE DONE AFTER JUST THREE COMPARISONS.                      INTERV33
C     OTHERWISE, WE REPEATEDLY DOUBLE THE DIFFERENCE  ISTEP = IHI - ILO INTERV34
C  WHILE ALSO MOVING  ILO  AND  IHI  IN THE DIRECTION OF  X , UNTIL     INTERV35
C                      XT(ILO) .LE. X .LT. XT(IHI) ,                    INTERV36
C  AFTER WHICH WE USE BISECTION TO GET, IN ADDITION, ILO+1 = IHI .      INTERV37
C  LEFT = ILO  IS THEN RETURNED.                                        INTERV38
C                                                                       INTERV39
      INTEGER LEFT,LXT,MFLAG,IHI,ILO,ISTEP,MIDDLE                       INTERV40
      REAL X,XT(LXT)                                                    INTERV41
      DATA ILO /1/                                                      INTERV42
C     SAVE ILO  (A VALID FORTRAN STATEMENT IN THE NEW 1977 STANDARD)    INTERV43
      IHI=ILO+1                                                         INTERV44
      IF (IHI.LT.LXT) GO TO 10                                          INTERV45
      IF (X.GE.XT(LXT)) GO TO 110                                       INTERV46
      IF (LXT.LE.1) GO TO 90                                            INTERV47
      ILO=LXT-1                                                         INTERV48
      IHI=LXT                                                           INTERV49
C                                                                       INTERV50
10    IF (X.GE.XT(IHI)) GO TO 40                                        INTERV51
      IF (X.GE.XT(ILO)) GO TO 100                                       INTERV52
C                                                                       INTERV53
C              **** NOW X .LT. XT(ILO) . DECREASE  ILO  TO CAPTURE  X . INTERV54
      ISTEP=1                                                           INTERV55
20    IHI=ILO                                                           INTERV56
      ILO=IHI-ISTEP                                                     INTERV57
```

```
      IF (ILO.LE.1) GO TO 30                                          INTERV58
      IF (X.GE.XT(ILO)) GO TO 70                                      INTERV59
      ISTEP=ISTEP*2                                                   INTERV60
      GO TO 20                                                        INTERV61
   30 ILO=1                                                           INTERV62
      IF (X.LT.XT(1)) GO TO 90                                        INTERV63
      GO TO 70                                                        INTERV64
C              **** NOW X .GE. XT(IHI) . INCREASE  IHI  TO CAPTURE  X . INTERV65
   40 ISTEP=1                                                         INTERV66
   50 ILO=IHI                                                         INTERV67
      IHI=ILO+ISTEP                                                   INTERV68
      IF (IHI.GE.LXT) GO TO 60                                        INTERV69
      IF (X.LT.XT(IHI)) GO TO 70                                      INTERV70
      ISTEP=ISTEP*2                                                   INTERV71
      GO TO 50                                                        INTERV72
   60 IF (X.GE.XT(LXT)) GO TO 110                                     INTERV73
      IHI=LXT                                                         INTERV74
C                                                                     INTERV75
C           **** NOW XT(ILO) .LE. X .LT. XT(IHI) . NARROW THE INTERVAL. INTERV76
   70 MIDDLE=(ILO+IHI)/2                                              INTERV77
      IF (MIDDLE.EQ.ILO) GO TO 100                                    INTERV78
C     NOTE. IT IS ASSUMED THAT MIDDLE = ILO IN CASE IHI = ILO+1 .     INTERV79
      IF (X.LT.XT(MIDDLE)) GO TO 80                                   INTERV80
      ILO=MIDDLE                                                      INTERV81
      GO TO 70                                                        INTERV82
   80 IHI=MIDDLE                                                      INTERV83
      GO TO 70                                                        INTERV84
C**** SET OUTPUT AND RETURN.                                           INTERV85
   90 MFLAG=-1                                                        INTERV86
      LEFT=1                                                          INTERV87
      RETURN                                                          INTERV88
  100 MFLAG=0                                                         INTERV89
      LEFT=ILO                                                        INTERV90
      RETURN                                                          INTERV91
  110 MFLAG=1                                                         INTERV92
      LEFT=LXT                                                        INTERV93
      RETURN                                                          INTERV94
      END                                                             INTERV95
```

CPR*NS(1).L2APPR(1)

```
      SUBROUTINE L2APPR (T,N,K,Q,DIAG,BCOEF,KMX,NPK,NTAU,TAU,GTAU,       L2APP001
     2 WEIGHT)                                                           L2APP002
C                                                                        L2APP003
      INTEGER N,K,KMX,NPK,NTAU                                           L2APP004
      REAL Q(KMX,N)                                                      L2APP005
      REAL T(NPK),DIAG(N),BCOEF(N),TAU(NTAU),GTAU(NTAU),WEIGHT(NTAU)     L2APP006
C                                                                        L2APP007
CONSTRUCTS THE (WEIGHTED DISCRETE) L2-APPROXIMATION BY SPLINES OF ORDER L2APP008
C  K  WITH KNOT SEQUENCE  T(1), ..., T(N+K)  TO GIVEN DATA POINTS        L2APP009
C  ( TAU(I), GTAU(I) ), I=1,...,NTAU. THE B-SPLINE COEFFICIENTS          L2APP010
C  B C O E F   OF THE APPROXIMATING SPLINE ARE DETERMINED FROM THE       L2APP011
C  NORMAL EQUATIONS USING CHOLESKY'S METHOD.                             L2APP012
C                                                                        L2APP013
C  ON INPUT.                                                             L2APP014
C                                                                        L2APP015
C     T(*)            IS AN ARRAY OF SIZE AT LEAST NPK = N + K AND HOLDS L2APP016
C                     THE KNOT SEQUENCE IN T(1) . . . T(NPK)             L2APP017
C                                                                        L2APP018
C     N               IS THE DIMENSION OF THE SPACE OF SPLINES OF ORDER KL2APP019
C                     WITH KNOTS T.                                      L2APP020
C                                                                        L2APP021
C     K               IS THE ORDER OF THE SPLINES = DEGREE + 1           L2APP022
C                                                                        L2APP023
C     Q(*, *)         IS A WORK ARRAY WITH ROW DIMENSION KMX AND COLUMN  L2APP024
C                     DIMENSION AT LEAST N.                              L2APP025
C                                                                        L2APP026
C     DIAG(*)         IS A WORK ARRAY OF SIZE AT LEAST N.                L2APP027
C                                                                        L2APP028
C     KMX             IS THE ROW DIMENSION OF Q.                         L2APP029
C                                                                        L2APP030
C     NPK             IS N + K.                                          L2APP031
C                                                                        L2APP032
C     NTAU            IS THE NUMBER OF DATA POINTS.                      L2APP033
C                                                                        L2APP034
C     TAU(*)          IS AN ARRAY OF SIZE AT LEAST NTAU WHICH CONTAINS   L2APP035
C                     THE ABSCISSAS OF THE DATA POINTS TO BE FITTED IN   L2APP036
C                     TAU(1) . . . TAU(NTAU).                            L2APP037
C                                                                        L2APP038
C     GTAU(*)         IS AN ARRAY OF SIZE AT LEAST NTAU WHICH CONTAINS   L2APP039
C                     THE ORDINATES OF THE DATA POINTS TO BE FITTED IN   L2APP040
C                     GTAU(1) . . . GTAU(NTAU).                          L2APP041
C                                                                        L2APP042
C     WEIGHT(*)       IS AN ARRAY OF SIZE AT LEAST NTAU WHICH CONTAINS   L2APP043
C                     THE CORRESPONDING WEIGHTS TO BE APPLIED TO THE     L2APP044
C                     DATA POINTS WHEN FITTING IN WEIGHT(1) . . .        L2APP045
C                     WEIGHT(NTAU).                                      L2APP046
C                                                                        L2APP047
C  ON OUTPUT.                                                            L2APP048
C                                                                        L2APP049
C     Q(*, *)         CONTAINS THE K LOWER DIAGONALS OF THE CHOLESKY     L2APP050
C                     FACTOR OF THE GRAMIAN MATRIX C IN ITS FIRST K ROWS.L2APP051
C                                                                        L2APP052
C     DIAG(*)         CONTAINS LITTLE OF IMPORTANCE.                     L2APP053
C                                                                        L2APP054
C     BCOEF(*)        IS AN ARRAY OF SIZE AT LEAST N WHICH CONTAINS THE NL2APP055
C                     B-SPLINE COEFFICIENTS OF THE L2 APPROXIMATION IN   L2APP056
C                     BCOEF(1) . . . BCOEF(N).                           L2APP057
```

```
C                                                                       L2APP058
C     AND THE REST OF THE VARIABLES ARE UNCHANGED.                      L2APP059
C                                                                       L2APP060
C******  M E T H O D  ******                                            L2APP061
C  THE B-SPLINE COEFFICIENTS OF THE L2-APPR. ARE DETERMINED AS THE SOL- L2APP062
C  UTION OF THE NORMAL EQUATIONS                                        L2APP063
C     SUM ( (B(I),B(J))*BCOEF(J) : J=1,...,N)  = (B(I),G),              L2APP064
C                                                I = 1, ..., N .        L2APP065
C  HERE,  B(I)  DENOTES THE I-TH B-SPLINE,  G  DENOTES THE FUNCTION TO  L2APP066
C  BE APPROXIMATED, AND THE  I N N E R   P R O D U C T  OF TWO FUNCT-   L2APP067
C  IONS  F  AND  G  IS GIVEN BY                                         L2APP068
C      (F,G)  :=  SUM ( F(TAU(I))*G(TAU(I))*WEIGHT(I) : I=1,...,NTAU) . L2APP069
C  THE ARRAYS  T A U  AND  W E I G H T  ARE GIVEN IN COMMON BLOCK       L2APP070
C   D A T A , AS IS THE ARRAY  G T A U  CONTAINING THE SEQUENCE         L2APP071
C  G(TAU(I)), I=1,...,NTAU.                                             L2APP072
C  THE RELEVANT FUNCTION VALUES OF THE B-SPLINES  B(I), I=1,...,N, ARE  L2APP073
C  SUPPLIED BY THE SUBPROGRAM  B S P L V B .                            L2APP074
C     THE COEFF.MATRIX  C , WITH                                        L2APP075
C           C(I,J)  :=  (B(I), B(J)), I,J=1,...,N,                      L2APP076
C  OF THE NORMAL EQUATIONS IS SYMMETRIC AND (2*K-1)-BANDED, THEREFORE   L2APP077
C  CAN BE SPECIFIED BY GIVING ITS K BANDS AT OR BELOW THE DIAGONAL. FOR L2APP078
C  I=1,...,N,  WE STORE                                                 L2APP079
C   (B(I),B(J))  =  C(I,J)  IN  Q(I-J+1,J), J=I,...,MIN0(I+K-1,N)       L2APP080
C  AND THE RIGHT SIDE                                                   L2APP081
C   (B(I), G )  IN  BCOEF(I) .                                          L2APP082
C  SINCE B-SPLINE VALUES ARE MOST EFFICIENTLY GENERATED BY FINDING SIM- L2APP083
C  ULTANEOUSLY THE VALUE OF  E V E R Y  NONZERO B-SPLINE AT ONE POINT,  L2APP084
C  THE ENTRIES OF  C  (I.E., OF  Q ), ARE GENERATED BY COMPUTING, FOR   L2APP085
C  EACH LL, ALL THE TERMS INVOLVING  TAU(LL)  SIMULTANEOUSLY AND ADDING L2APP086
C  THEM TO ALL RELEVANT ENTRIES.                                        L2APP087
C                                                                       L2APP088
C  ADDITIONAL ROUTINES REQUIRED.                                        L2APP089
C                                                                       L2APP090
C     BSPLVB   BCHFAC   BCHSLV                                          L2APP091
C                                                                       L2APP092
C  MODIFICATION BY.                                                     L2APP093
C                                                                       L2APP094
C     MARTIN CORDES                                                     L2APP095
C     CENTER FOR APPLIED MATHEMATICS, NBS                               L2APP096
C     VERSION 1                                                         L2APP097
C     OCT 1979                                                          L2APP098
C                                                                       L2APP099
C-----------------------------------------------------------------------L2APP100
C                                                                       L2APP101
      REAL BIATX(20)                                                    L2APP102
      REAL DW                                                           L2APP103
      INTEGER I,J,JJ,LEFT,LEFTMK,LL,MM                                  L2APP104
10    FORMAT (/5X,5H<<<<<I4,1X,22HB-SPLINE COEFFICIENTS ,               L2APP105
     2 14HCOMPUTED >>>>>)                                               L2APP106
C                                                                       L2APP107
      DO 20 J=1,N                                                       L2APP108
      BCOEF(J)=0.                                                       L2APP109
      DO 20 I=1,K                                                       L2APP110
20    Q(I,J)=0.                                                         L2APP111
      LEFT=K                                                            L2APP112
      LEFTMK=0                                                          L2APP113
      DO 50 LL=1,NTAU                                                   L2APP114
C                  LOCATE  L E F T  S.T. TAU(LL) IN (T(LEFT),T(LEFT+1))L2APP115
```

```
   30 IF (LEFT.EQ.N) GO TO 40                                            L2APP116
      IF (TAU(LL).LT.T(LEFT+1)) GO TO 40                                 L2APP117
      LEFT=LEFT+1                                                        L2APP118
      LEFTMK=LEFTMK+1                                                    L2APP119
      GO TO 30                                                           L2APP120
   40 CALL BSPLVB (T,K,1,TAU(LL),LEFT,BIATX)                             L2APP121
C        BIATX(MM) CONTAINS THE VALUE OF B(LEFT-K+MM) AT TAU(LL).        L2APP122
C        HENCE, WITH  DW := BIATX(MM)*WEIGHT(LL), THE NUMBER DW*GTAU(LL)L2APP123
C        IS A SUMMAND IN THE INNER PRODUCT                               L2APP124
C           (B(LEFT-K+MM), G)  WHICH GOES INTO  BCOEF(LEFT-K+MM)        L2APP125
C        AND THE NUMBER BIATX(JJ)*DW IS A SUMMAND IN THE INNER PRODUCT  L2APP126
C           (B(LEFT-K+JJ), B(LEFT-K+MM)), INTO  Q(JJ-MM+1,LEFT-K+MM)    L2APP127
C        SINCE  (LEFT-K+JJ) - (LEFT-K+MM) + 1  =  JJ - MM + 1 .          L2APP128
      DO 50 MM=1,K                                                       L2APP129
      DW=BIATX(MM)*WEIGHT(LL)                                            L2APP130
      J=LEFTMK+MM                                                        L2APP131
      BCOEF(J)=DW*GTAU(LL)+BCOEF(J)                                      L2APP132
      I=1                                                                L2APP133
      DO 50 JJ=MM,K                                                      L2APP134
      Q(I,J)=BIATX(JJ)*DW+Q(I,J)                                         L2APP135
   50 I=I+1                                                              L2APP136
C                                                                        L2APP137
C             CONSTRUCT CHOLESKY FACTORIZATION FOR  C  IN  Q , THEN USE L2APP138
C             IT TO SOLVE THE NORMAL EQUATIONS                           L2APP139
C                    C*X  =  BCOEF                                       L2APP140
C             FOR  X , AND STORE  X  IN  BCOEF .                         L2APP141
      CALL BCHFAC (Q,KMX,K,N,DIAG)                                       L2APP142
      CALL BCHSLV (Q,KMX,K,N,BCOEF)                                      L2APP143
      WRITE (6,10) N                                                     L2APP144
      RETURN                                                             L2APP145
      END                                                                L2APP146
```

CPR*NS(1).NORCDF(1)

```
      SUBROUTINE NORCDF (X,CDF)                                         NORCDF01
C                                                                       NORCDF02
C     PURPOSE--THIS SUBROUTINE COMPUTES THE CUMULATIVE DISTRIBUTION     NORCDF03
C              FUNCTION VALUE FOR THE NORMAL (GAUSSIAN)                 NORCDF04
C              DISTRIBUTION WITH MEAN = 0 AND STANDARD DEVIATION = 1.   NORCDF05
C              THIS DISTRIBUTION IS DEFINED FOR ALL X AND HAS           NORCDF06
C              THE PROBABILITY DENSITY FUNCTION                         NORCDF07
C              F(X) = (1/SQRT(2*PI))*EXP(-X*X/2).                       NORCDF08
C     INPUT  ARGUMENTS--X      = THE SINGLE PRECISION VALUE AT          NORCDF09
C                                WHICH THE CUMULATIVE DISTRIBUTION      NORCDF10
C                                FUNCTION IS TO BE EVALUATED.           NORCDF11
C     OUTPUT ARGUMENTS--CDF    = THE SINGLE PRECISION CUMULATIVE        NORCDF12
C                                DISTRIBUTION FUNCTION VALUE.           NORCDF13
C     OUTPUT--THE SINGLE PRECISION CUMULATIVE DISTRIBUTION              NORCDF14
C             FUNCTION VALUE CDF.                                       NORCDF15
C     PRINTING--NONE.                                                   NORCDF16
C     RESTRICTIONS--NONE.                                               NORCDF17
C     OTHER DATAPAC   SUBROUTINES NEEDED--NONE.                         NORCDF18
C     FORTRAN LIBRARY SUBROUTINES NEEDED--EXP.                          NORCDF19
C     MODE OF INTERNAL OPERATIONS--SINGLE PRECISION.                    NORCDF20
C     LANGUAGE--ANSI FORTRAN.                                           NORCDF21
C     REFERENCES--NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS      NORCDF22
C                 SERIES 55, 1964, PAGE 932, FORMULA 26.2.17.           NORCDF23
C               --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE               NORCDF24
C                 DISTRIBUTIONS--1, 1970, PAGES 40-111.                 NORCDF25
C     WRITTEN BY--JAMES J. FILLIBEN                                     NORCDF26
C                 STATISTICAL ENGINEERING LABORATORY (205.03)           NORCDF27
C                 NATIONAL BUREAU OF STANDARDS                          NORCDF28
C                 WASHINGTON, D. C. 20234                               NORCDF29
C                 PHONE:  301-921-2315                                  NORCDF30
C     ORIGINAL VERSION--JUNE      1972.                                 NORCDF31
C     UPDATED         --SEPTEMBER 1975.                                 NORCDF32
C     UPDATED         --NOVEMBER  1975.                                 NORCDF33
C                                                                       NORCDF34
C-----------------------------------------------------------------------NORCDF35
C                                                                       NORCDF36
      DATA B1,B2,B3,B4,B5,P /.319381530,-0.356563782,1.781477937,       NORCDF37
     2 -1.821255978,1.330274429,.2316419/                               NORCDF38
C                                                                       NORCDF39
      IPR=6                                                             NORCDF40
C                                                                       NORCDF41
C     CHECK THE INPUT ARGUMENTS FOR ERRORS.                             NORCDF42
C     NO INPUT ARGUMENT ERRORS POSSIBLE                                 NORCDF43
C     FOR THIS DISTRIBUTION.                                            NORCDF44
C                                                                       NORCDF45
C-----START POINT-------------------------------------------------------NORCDF46
C                                                                       NORCDF47
      Z=X                                                               NORCDF48
      IF (X.LT.0.0) Z=-Z                                                NORCDF49
      T=1.0/(1.0+P*Z)                                                   NORCDF50
      CDF=1.0-((0.39894228040143)*EXP(-0.5*Z*Z))*(B1*T+B2*T**2+B3*T**3+BNORCDF51
     24*T**4+B5*T**5)                                                   NORCDF52
      IF (X.LT.0.0) CDF=1.0-CDF                                         NORCDF53
C                                                                       NORCDF54
      RETURN                                                            NORCDF55
      END                                                               NORCDF56
```

CPR*NS(1).NORPPF(1)

```
      SUBROUTINE NORPPF (P,PPF)                                    NORPP001
C                                                                  NORPP002
C     PURPOSE--THIS SUBROUTINE COMPUTES THE PERCENT POINT          NORPP003
C              FUNCTION VALUE FOR THE NORMAL (GAUSSIAN)            NORPP004
C              DISTRIBUTION WITH MEAN = 0 AND STANDARD DEVIATION = 1.  NORPP005
C              THIS DISTRIBUTION IS DEFINED FOR ALL X AND HAS      NORPP006
C              THE PROBABILITY DENSITY FUNCTION                    NORPP007
C              F(X) = (1/SQRT(2*PI))*EXP(-X*X/2).                  NORPP008
C              NOTE THAT THE PERCENT POINT FUNCTION OF A DISTRIBUTION  NORPP009
C              IS IDENTICALLY THE SAME AS THE INVERSE CUMULATIVE   NORPP010
C              DISTRIBUTION FUNCTION OF THE DISTRIBUTION.          NORPP011
C     INPUT  ARGUMENTS--P      = THE SINGLE PRECISION VALUE        NORPP012
C                                (BETWEEN 0.0 AND 1.0)             NORPP013
C                                AT WHICH THE PERCENT POINT        NORPP014
C                                FUNCTION IS TO BE EVALUATED.      NORPP015
C     OUTPUT ARGUMENTS--PPF    = THE SINGLE PRECISION PERCENT      NORPP016
C                                POINT FUNCTION VALUE.             NORPP017
C     OUTPUT--THE SINGLE PRECISION PERCENT POINT                   NORPP018
C             FUNCTION VALUE PPF.                                  NORPP019
C     PRINTING--NONE UNLESS AN INPUT ARGUMENT ERROR CONDITION EXISTS.  NORPP020
C     RESTRICTIONS--P SHOULD BE BETWEEN 0.0 AND 1.0, EXCLUSIVELY.  NORPP021
C     OTHER DATAPAC   SUBROUTINES NEEDED--NONE.                    NORPP022
C     FORTRAN LIBRARY SUBROUTINES NEEDED--SQRT, ALOG.              NORPP023
C     MODE OF INTERNAL OPERATIONS--SINGLE PRECISION.               NORPP024
C     LANGUAGE--ANSI FORTRAN.                                      NORPP025
C     REFERENCES--ODEH AND EVANS, THE PERCENTAGE POINTS            NORPP026
C                   OF THE NORMAL DISTRIBUTION, ALGORTIHM 70,      NORPP027
C                   APPLIED STATISTICS, 1974, PAGES 96-97.         NORPP028
C                 --EVANS, ALGORITHMS FOR MINIMAL DEGREE           NORPP029
C                   POLYNOMIAL AND RATIONAL APPROXIMATION,         NORPP030
C                   M. SC. THESIS, 1972, UNIVERSITY                NORPP031
C                   OF VICTORIA, B. C., CANADA.                    NORPP032
C                 --HASTINGS, APPROXIMATIONS FOR DIGITAL           NORPP033
C                   COMPUTERS, 1955, PAGES 113, 191, 192.          NORPP034
C                 --NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS  NORPP035
C                   SERIES 55, 1964, PAGE 933, FORMULA 26.2.23.    NORPP036
C                 --FILLIBEN, SIMPLE AND ROBUST LINEAR ESTIMATION  NORPP037
C                   OF THE LOCATION PARAMETER OF A SYMMETRIC       NORPP038
C                   DISTRIBUTION (UNPUBLISHED PH.D. DISSERTATION,  NORPP039
C                   PRINCETON UNIVERSITY), 1969, PAGES 21-44, 229-231.  NORPP040
C                 --FILLIBEN, 'THE PERCENT POINT FUNCTION',        NORPP041
C                   (UNPUBLISHED MANUSCRIPT), 1970, PAGES 28-31.   NORPP042
C                 --JOHNSON AND KOTZ, CONTINUOUS UNIVARIATE        NORPP043
C                   DISTRIBUTIONS--1, 1970, PAGES 40-111.          NORPP044
C                 --THE KELLEY STATISTICAL TABLES, 1948.           NORPP045
C                 --OWEN, HANDBOOK OF STATISTICAL TABLES,          NORPP046
C                   1962, PAGES 3-16.                              NORPP047
C                 --PEARSON AND HARTLEY, BIOMETRIKA TABLES         NORPP048
C                   FOR STATISTICIANS, VOLUME 1, 1954,             NORPP049
C                   PAGES 104-113.                                 NORPP050
C     COMMENTS--THE CODING AS PRESENTED BELOW                      NORPP051
C               IS ESSENTIALLY IDENTICAL TO THAT                   NORPP052
C               PRESENTED BY ODEH AND EVANS                        NORPP053
C               AS ALGORTIHM 70 OF APPLIED STATISTICS.             NORPP054
C               THE PRESENT AUTHOR HAS MODIFIED THE                NORPP055
C               ORIGINAL ODEH AND EVANS CODE WITH ONLY             NORPP056
C               MINOR STYLISTIC CHANGES.                           NORPP057
```

```
C              --AS POINTED OUT BY ODEH AND EVANS                         NORPP058
C                IN APPLIED STATISTICS,                                   NORPP059
C                THEIR ALGORITHM REPRESENTES A                            NORPP060
C                SUBSTANTIAL IMPROVEMENT OVER THE                         NORPP061
C                PREVIOUSLY EMPLOYED                                      NORPP062
C                HASTINGS APPROXIMATION FOR THE                           NORPP063
C                NORMAL PERCENT POINT FUNCTION--                          NORPP064
C                THE ACCURACY OF APPROXIMATION                            NORPP065
C                BEING IMPROVED FROM 4.5*(10**-4)                         NORPP066
C                TO 1.5*(10**-8).                                         NORPP067
C     WRITTEN BY--JAMES J. FILLIBEN                                       NORPP068
C                 STATISTICAL ENGINEERING LABORATORY (205.03)             NORPP069
C                 NATIONAL BUREAU OF STANDARDS                            NORPP070
C                 WASHINGTON, D. C. 20234                                 NORPP071
C                 PHONE:  301-921-2315                                    NORPP072
C     ORIGINAL VERSION--JUNE      1972.                                   NORPP073
C     UPDATED         --SEPTEMBER 1975.                                   NORPP074
C     UPDATED         --NOVEMBER  1975.                                   NORPP075
C     UPDATED         --OCTOBER   1976.                                   NORPP076
C                                                                         NORPP077
C-----------------------------------------------------------------------  NORPP078
C                                                                         NORPP079
      DATA P0,P1,P2,P3,P4 /-.322232431088,-1.0,-.342242088547,            NORPP080
     2 -.204231210245E-1,-.453642210148E-4/                               NORPP081
      DATA Q0,Q1,Q2,Q3,Q4 /.993484626060E-1,.588581570495,.531103462366,NORPP082
     2 .103537752850,.38560700634E-2/                                     NORPP083
C                                                                         NORPP084
      IPR=6                                                               NORPP085
C                                                                         NORPP086
C     CHECK THE INPUT ARGUMENTS FOR ERRORS                                NORPP087
C                                                                         NORPP088
      IF (P.LE.0.0.OR.P.GE.1.0) GO TO 10                                  NORPP089
      GO TO 20                                                            NORPP090
10    WRITE (IPR,30)                                                      NORPP091
      WRITE (IPR,40) P                                                    NORPP092
      RETURN                                                              NORPP093
20    CONTINUE                                                            NORPP094
30    FORMAT (1H ,115H***** FATAL ERROR--THE FIRST  INPUT ARGUMENT TO THNORPP095
     2E NORPPF SUBROUTINE IS OUTSIDE THE ALLOWABLE (0,1) INTERVAL *****)NORPP096
40    FORMAT (1H ,35H***** THE VALUE OF THE ARGUMENT IS ,E15.8,6H *****)NORPP097
C                                                                         NORPP098
C-----START POINT-------------------------------------------------------  NORPP099
C                                                                         NORPP100
      IF (P.NE.0.5) GO TO 50                                              NORPP101
      PPF=0.0                                                             NORPP102
      RETURN                                                              NORPP103
C                                                                         NORPP104
50    R=P                                                                 NORPP105
      IF (P.GT.0.5) R=1.0-R                                               NORPP106
      T=SQRT(-2.0*ALOG(R))                                                NORPP107
      ANUM=((((T*P4+P3)*T+P2)*T+P1)*T+P0)                                 NORPP108
      ADEN=((((T*Q4+Q3)*T+Q2)*T+Q1)*T+Q0)                                 NORPP109
      PPF=T+(ANUM/ADEN)                                                   NORPP110
      IF (P.LT.0.5) PPF=-PPF                                              NORPP111
      RETURN                                                              NORPP112
C                                                                         NORPP113
      END                                                                 NORPP114
```

CPR*NS(1).PLOTC(2)

```
         SUBROUTINE PLOTC (Y,X,CHAR,N,ITYPE)                             PLOTC001
C                                                                        PLOTC002
C        PURPOSE--THIS SUBROUTINE YIELDS A ONE-PAGE PRINTER PLOT         PLOTC003
C                 OF Y(I) VERSUS X(I) WITH SPECIAL PLOTTING              PLOTC004
C                 CHARACTERS.                                            PLOTC005
C                 THIS 'SPECIAL PLOTTING CHARACTER' CAPABILITY           PLOTC006
C                 ALLOWS THE DATA ANALYST TO INCORPORATE INFORMATION     PLOTC007
C                 FROM A THIRD VARIABLE (ASIDE FROM Y AND X) INTO        PLOTC008
C                 THE PLOT.                                              PLOTC009
C                 THE PLOT CHARACTER USED AT THE I-TH PLOTTING           PLOTC010
C                 POSITION (THAT IS, AT THE COORDINATE (X(I),Y(I)))      PLOTC011
C                 WILL BE                                                PLOTC012
C                 1 IF CHAR(I) IS BETWEEN  0.5 AND  1.5                  PLOTC013
C                 2 IF CHAR(I) IS BETWEEN  1.5 AND  2.5                  PLOTC014
C                   .                                                    PLOTC015
C                   .                                                    PLOTC016
C                   .                                                    PLOTC017
C                 9 IF CHAR(I) IS BETWEEN  8.5 AND  9.5                  PLOTC018
C                 . IF CHAR(I) IS BETWEEN  9.5 AND 10.5                  PLOTC019
C                 A IF CHAR(I) IS BETWEEN 10.5 AND 11.5                  PLOTC020
C                 B IF CHAR(I) IS BETWEEN 11.5 AND 12.5                  PLOTC021
C                 C IF CHAR(I) IS BETWEEN 12.5 AND 13.5                  PLOTC022
C                   .                                                    PLOTC023
C                   .                                                    PLOTC024
C                   .                                                    PLOTC025
C                 W IF CHAR(I) IS BETWEEN 32.5 AND 33.5                  PLOTC026
C                 X IF CHAR(I) IS BETWEEN 33.5 AND 34.5                  PLOTC027
C                 Y IF CHAR(I) IS BETWEEN 34.5 AND 35.5                  PLOTC028
C                 Z IF CHAR(I) IS BETWEEN 35.5 AND 36.5                  PLOTC029
C                 X IF CHAR(I) IS ANY VALUE OUTSIDE THE RANGE            PLOTC030
C                                            0.5 TO  36.5.               PLOTC031
C        INPUT ARGUMENTS--Y      = THE SINGLE PRECISION VECTOR OF        PLOTC032
C                                  (UNSORTED OR SORTED) OBSERVATIONS     PLOTC033
C                                  TO BE PLOTTED VERTICALLY.             PLOTC034
C                       --X      = THE SINGLE PRECISION VECTOR OF        PLOTC035
C                                  (UNSORTED OR SORTED) OBSERVATIONS     PLOTC036
C                                  TO BE PLOTTED HORIZONTALLY.           PLOTC037
C                       --CHAR   = THE SINGLE PRECISION VECTOR OF        PLOTC038
C                                  OBSERVATIONS WHICH CONTROL THE        PLOTC039
C                                  VALUE OF EACH INDIVIDUAL PLOT         PLOTC040
C                                  CHARACTER.                            PLOTC041
C                       --N      = THE INTEGER NUMBER OF OBSERVATIONS    PLOTC042
C                                  IN THE VECTOR Y.                      PLOTC043
C        OUTPUT--A ONE-PAGE PRINTER PLOT OF Y(I) VERSUS X(I)             PLOTC044
C                WITH SPECIAL PLOT CHARACTERS.                           PLOTC045
C        PRINTING--YES.                                                  PLOTC046
C        RESTRICTIONS--THERE IS NO RESTRICTION ON THE MAXIMUM VALUE      PLOTC047
C                      OF N FOR THIS SUBROUTINE.                         PLOTC048
C        OTHER DATAPAC   SUBROUTINES NEEDED--NONE.                       PLOTC049
C        FORTRAN LIBRARY SUBROUTINES NEEDED--NONE.                       PLOTC050
C        MODE OF INTERNAL OPERATIONS--SINGLE PRECISION.                  PLOTC051
C        LANGUAGE--ANSI FORTRAN.                                         PLOTC052
C        COMMENT--VALUES IN THE VERTICAL AXIS VECTOR (Y),                PLOTC053
C                 THE HORIZONTAL AXIS VECTOR (X),                        PLOTC054
C                 OR THE PLOT CHARACTER VECTOR (CHAR) WHICH ARE          PLOTC055
C                 EQUAL TO OR IN EXCESS OF 10.0**10 WILL NOT BE          PLOTC056
C                 PLOTTED.                                               PLOTC057
```

```
C               THIS CONVENTION GREATLY SIMPLIFIES THE PROBLEM         PLOTC058
C               OF PLOTTING WHEN SOME ELEMENTS IN THE VECTOR Y         PLOTC059
C               (OR X, OR CHAR) ARE 'MISSING DATA', OR WHEN WE PURPOSELY PLOTC060
C               WANT TO IGNORE CERTAIN ELEMENTS IN THE VECTOR Y        PLOTC061
C               (OR X, OR CHAR) FOR PLOTTING PURPOSES (THAT IS, WE DO NOTPLOTC062
C               WANT CERTAIN ELEMENTS IN Y (OR X, OR CHAR) TO BE       PLOTC063
C               PLOTTED).                                              PLOTC064
C               TO CAUSE SPECIFIC ELEMENTS IN Y (OR X, OR CHAR) TO BE  PLOTC065
C               IGNORED, WE REPLACE THE ELEMENTS BEFOREHAND            PLOTC066
C               (BY, FOR EXAMPLE, USE OF THE   REPLAC   SUBROUTINE)    PLOTC067
C               BY SOME LARGE VALUE (LIKE, SAY, 10.0**10) AND          PLOTC068
C               THEY WILL SUBSEQUENTLY BE IGNORED IN THE PLOTC         PLOTC069
C               SUBROUTINE.                                            PLOTC070
C     REFERENCES--FILLIBEN, 'STATISTICAL ANALYSIS OF INTERLAB          PLOTC071
C                 FATIGUE TIME DATA', UNPUBLISHED MANUSCRIPT           PLOTC072
C                 (AVAILABLE FROM AUTHOR)                              PLOTC073
C                 PRESENTED AT THE 'COMPUTER-ASSISTED DATA             PLOTC074
C                 ANALYSIS' SESSION AT THE NATIONAL MEETING            PLOTC075
C                 OF THE AMERICAN STATISTICAL ASSOCIATION,             PLOTC076
C                 NEW YORK CITY, DECEMBER 27-30, 1973.                 PLOTC077
C     WRITTEN BY--JAMES J. FILLIBEN                                    PLOTC078
C                 STATISTICAL ENGINEERING LABORATORY (205.03)          PLOTC079
C                 NATIONAL BUREAU OF STANDARDS                         PLOTC080
C                 WASHINGTON, D. C. 20234                              PLOTC081
C                 PHONE--301-921-2315                                  PLOTC082
C     ORIGINAL VERSION--OCTOBER   1974.                                PLOTC083
C     UPDATED         --NOVEMBER  1974.                                PLOTC084
C     UPDATED         --JANUARY   1975.                                PLOTC085
C     UPDATED         --JULY      1975.                                PLOTC086
C     UPDATED         --SEPTEMBER 1975.                                PLOTC087
C     UPDATED         --OCTOBER   1975.                                PLOTC088
C     UPDATED         --NOVEMBER  1975.                                PLOTC089
C     UPDATED         --FEBRUARY  1976.                                PLOTC090
C     UPDATED         --FEBRUARY  1977.                                PLOTC091
C     MINOR UPDATES   --APRIL     1980   BY CHARLES P. REEVE.          PLOTC092
C                                                                      PLOTC093
C-----------------------------------------------------------------     PLOTC094
C                                                                      PLOTC095
      INTEGER SBNAME(6)                                                PLOTC096
      INTEGER ALPHA1(6),ALPHA2(6),ALPHA3(6),ALPHA4(6)                  PLOTC097
      INTEGER BLANK,HYPHEN,ALPHAI,ALPHAX                               PLOTC098
      INTEGER ALPHAM,ALPHAA,ALPHAD,ALPHAN,EQUAL                        PLOTC099
      DIMENSION Y(1),X(1),CHAR(1),YLABLE(11),IPLOTC(37)                PLOTC100
      COMMON /BLOCK1/ IGRAPH(55,130)                                   PLOTC101
C                                                                      PLOTC102
      DATA SBNAME(1),SBNAME(2),SBNAME(3),SBNAME(4),SBNAME(5),SBNAME(6), PLOTC103
     2 ALPHA1(1),ALPHA1(2),ALPHA1(3),ALPHA1(4),ALPHA1(5),ALPHA1(6),    PLOTC104
     3 ALPHA2(1),ALPHA2(2),ALPHA2(3),ALPHA2(4),ALPHA2(5),ALPHA2(6),    PLOTC105
     4 ALPHA3(1),ALPHA3(2),ALPHA3(3),ALPHA3(4),ALPHA3(5),ALPHA3(6),    PLOTC106
     5 ALPHA4(1),ALPHA4(2),ALPHA4(3),ALPHA4(4),ALPHA4(5),ALPHA4(6) /1HP,PLOTC107
     6 1HL,1HO,1HT,1HC,1H ,1HF,1HI,1HR,1HS,1HT,1H ,1HS,1HE,1HC,1HO,1HN, PLOTC108
     7 1HD,1HT,1HH,1HI,1HR,1HD,1H ,1HF,1HO,1HU,1HR,1HT,1HH/            PLOTC109
      DATA BLANK,HYPHEN,ALPHAI,ALPHAX /1H ,1H-,1HI,1HX/                PLOTC110
      DATA ALPHAM,ALPHAA,ALPHAD,ALPHAN,EQUAL /1HM,1HA,1HD,1HN,1H=/     PLOTC111
      DATA IPLOTC(1),IPLOTC(2),IPLOTC(3),IPLOTC(4),IPLOTC(5),IPLOTC(6), PLOTC112
     2 IPLOTC(7),IPLOTC(8),IPLOTC(9),IPLOTC(10),IPLOTC(11),IPLOTC(12),  PLOTC113
     3 IPLOTC(13),IPLOTC(14),IPLOTC(15),IPLOTC(16),IPLOTC(17),IPLOTC(18)PLOTC114
     4,IPLOTC(19),IPLOTC(20),IPLOTC(21),IPLOTC(22),IPLOTC(23),IPLOTC(24)PLOTC115
```

```
     5,IPLOTC(25),IPLOTC(26),IPLOTC(27),IPLOTC(28),IPLOTC(29),IPLOTC(30)PLOTC116
     6,IPLOTC(31),IPLOTC(32),IPLOTC(33),IPLOTC(34),IPLOTC(35),IPLOTC(36)PLOTC117
     7,IPLOTC(37) /1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H.,1HA,1HB,1HC, PLOTC118
     8 1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS, PLOTC119
     9 1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1HX/                                 PLOTC120
C                                                                       PLOTC121
      IPR=6                                                             PLOTC122
      CUTOFF=(10.0**10)-1000.0                                          PLOTC123
C                                                                       PLOTC124
C     CHECK THE INPUT ARGUMENTS FOR ERRORS                              PLOTC125
C                                                                       PLOTC126
      IF (ITYPE.EQ.1) WRITE (IPR,520)                                   PLOTC127
      IF (ITYPE.EQ.2) WRITE (IPR,530)                                   PLOTC128
      IF (N.LT.1) GO TO 10                                              PLOTC129
      GO TO 20                                                          PLOTC130
   10 WRITE (IPR,200)                                                   PLOTC131
      WRITE (IPR,210)                                                   PLOTC132
      WRITE (IPR,230) (ALPHA4(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC133
      WRITE (IPR,260) N                                                 PLOTC134
      WRITE (IPR,200)                                                   PLOTC135
      RETURN                                                            PLOTC136
   20 CONTINUE                                                          PLOTC137
      IF (N.EQ.1) GO TO 30                                              PLOTC138
      GO TO 40                                                          PLOTC139
   30 WRITE (IPR,200)                                                   PLOTC140
      WRITE (IPR,210)                                                   PLOTC141
      WRITE (IPR,230) (ALPHA4(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC142
      WRITE (IPR,270) N                                                 PLOTC143
      WRITE (IPR,200)                                                   PLOTC144
      RETURN                                                            PLOTC145
   40 CONTINUE                                                          PLOTC146
C                                                                       PLOTC147
      HOLD=Y(1)                                                         PLOTC148
      DO 50 I=2,N                                                       PLOTC149
      IF (Y(I).NE.HOLD) GO TO 60                                        PLOTC150
   50 CONTINUE                                                          PLOTC151
      WRITE (IPR,200)                                                   PLOTC152
      WRITE (IPR,210)                                                   PLOTC153
      WRITE (IPR,230) (ALPHA1(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC154
      WRITE (IPR,280) HOLD                                              PLOTC155
      WRITE (IPR,200)                                                   PLOTC156
      RETURN                                                            PLOTC157
   60 CONTINUE                                                          PLOTC158
      HOLD=X(1)                                                         PLOTC159
      DO 70 I=2,N                                                       PLOTC160
      IF (X(I).NE.HOLD) GO TO 80                                        PLOTC161
   70 CONTINUE                                                          PLOTC162
      WRITE (IPR,200)                                                   PLOTC163
      WRITE (IPR,210)                                                   PLOTC164
      WRITE (IPR,230) (ALPHA2(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC165
      WRITE (IPR,280) HOLD                                              PLOTC166
      WRITE (IPR,200)                                                   PLOTC167
      RETURN                                                            PLOTC168
   80 CONTINUE                                                          PLOTC169
      HOLD=CHAR(1)                                                      PLOTC170
      DO 90 I=2,N                                                       PLOTC171
      IF (CHAR(I).NE.HOLD) GO TO 100                                    PLOTC172
   90 CONTINUE                                                          PLOTC173
```

```
      WRITE (IPR,200)                                                   PLOTC174
      WRITE (IPR,220)                                                   PLOTC175
      WRITE (IPR,230) (ALPHA3(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC176
      WRITE (IPR,280) HOLD                                              PLOTC177
      WRITE (IPR,200)                                                   PLOTC178
  100 CONTINUE                                                          PLOTC179
C                                                                       PLOTC180
      DO 110 I=1,N                                                      PLOTC181
      IF (Y(I).LT.CUTOFF) GO TO 120                                     PLOTC182
  110 CONTINUE                                                          PLOTC183
      WRITE (IPR,200)                                                   PLOTC184
      WRITE (IPR,210)                                                   PLOTC185
      WRITE (IPR,230) (ALPHA1(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC186
      WRITE (IPR,290)                                                   PLOTC187
      WRITE (IPR,300) CUTOFF                                            PLOTC188
      WRITE (IPR,200)                                                   PLOTC189
      RETURN                                                            PLOTC190
  120 CONTINUE                                                          PLOTC191
      DO 130 I=1,N                                                      PLOTC192
      IF (X(I).LT.CUTOFF) GO TO 140                                     PLOTC193
  130 CONTINUE                                                          PLOTC194
      WRITE (IPR,200)                                                   PLOTC195
      WRITE (IPR,210)                                                   PLOTC196
      WRITE (IPR,230) (ALPHA2(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC197
      WRITE (IPR,290)                                                   PLOTC198
      WRITE (IPR,300) CUTOFF                                            PLOTC199
      WRITE (IPR,200)                                                   PLOTC200
      RETURN                                                            PLOTC201
  140 CONTINUE                                                          PLOTC202
      DO 150 I=1,N                                                      PLOTC203
      IF (CHAR(I).LT.CUTOFF) GO TO 160                                  PLOTC204
  150 CONTINUE                                                          PLOTC205
      WRITE (IPR,200)                                                   PLOTC206
      WRITE (IPR,210)                                                   PLOTC207
      WRITE (IPR,230) (ALPHA3(L),L=1,6),(SBNAME(L),L=1,6)               PLOTC208
      WRITE (IPR,290)                                                   PLOTC209
      WRITE (IPR,300) CUTOFF                                            PLOTC210
      WRITE (IPR,200)                                                   PLOTC211
      RETURN                                                            PLOTC212
  160 CONTINUE                                                          PLOTC213
C                                                                       PLOTC214
      N2=0                                                              PLOTC215
      DO 180 I=1,N                                                      PLOTC216
      IF (Y(I).LT.CUTOFF.AND.X(I).LT.CUTOFF.AND.CHAR(I).LT.CUTOFF) GO TOPLOTC217
     2 170                                                              PLOTC218
      GO TO 180                                                         PLOTC219
  170 N2=N2+1                                                           PLOTC220
      IF (N2.GE.2) GO TO 190                                            PLOTC221
  180 CONTINUE                                                          PLOTC222
      WRITE (IPR,200)                                                   PLOTC223
      WRITE (IPR,210)                                                   PLOTC224
      WRITE (IPR,240) (ALPHA1(L),L=1,6),(ALPHA2(L),L=1,6),(ALPHA3(L),L=1PLOTC225
     2,6)                                                               PLOTC226
      WRITE (IPR,250) (SBNAME(L),L=1,6)                                 PLOTC227
      WRITE (IPR,310)                                                   PLOTC228
      WRITE (IPR,320) N2                                                PLOTC229
      WRITE (IPR,200)                                                   PLOTC230
      RETURN                                                            PLOTC231
```

```
 190  CONTINUE                                                           PLOTC232
C                                                                        PLOTC233
 200  FORMAT (1H ,50H**************************************************,PLOTC234
     2 20H********************)                                          PLOTC235
 210  FORMAT (1H ,50H                     FATAL ERROR                  )PLOTC236
 220  FORMAT (1H ,50H                  NON-FATAL DIAGNOSTIC            )PLOTC237
 230  FORMAT (1H ,4HTHE ,6A1,23H INPUT ARGUMENT TO THE ,6A1,             PLOTC238
     2 11H SUBROUTINE)                                                   PLOTC239
 240  FORMAT (1H ,4HTHE ,6A1,2H, ,6A1,6H, AND ,6A1)                      PLOTC240
 250  FORMAT (1H ,23HINPUT ARGUMENTS TO THE ,6A1,11H SUBROUTINE)         PLOTC241
 260  FORMAT (1H ,30HIS NON-NEGATIVE (WITH VALUE = ,I8,1H))              PLOTC242
 270  FORMAT (1H ,15HHAS THE VALUE 1)                                    PLOTC243
 280  FORMAT (1H ,19HHAS ALL ELEMENTS = ,E15.8)                          PLOTC244
 290  FORMAT (1H ,40HHAS ALL ELEMENTS IN EXCESS OF THE CUTOFF)           PLOTC245
 300  FORMAT (1H ,9HVALUE OF ,E15.8)                                     PLOTC246
 310  FORMAT (1H ,39HARE SUCH THAT TOO MANY POINTS HAVE BEEN,            PLOTC247
     2 24H EXCLUDED FROM THE PLOT.)                                      PLOTC248
 320  FORMAT (1H ,5HONLY ,I3,31H POINTS ARE LEFT TO BE PLOTTED.)         PLOTC249
C                                                                        PLOTC250
C-----START POINT------------------------------------------------------- PLOTC251
C                                                                        PLOTC252
C     DETERMINE THE VALUES TO BE LISTED ON THE LEFT VERTICAL AXIS        PLOTC253
C                                                                        PLOTC254
      DO 330 I=1,N                                                       PLOTC255
      IF (Y(I).GE.CUTOFF) GO TO 330                                      PLOTC256
      IF (X(I).GE.CUTOFF) GO TO 330                                      PLOTC257
      IF (CHAR(I).GE.CUTOFF) GO TO 330                                   PLOTC258
      YMIN=Y(I)                                                          PLOTC259
      YMAX=Y(I)                                                          PLOTC260
      GO TO 340                                                          PLOTC261
 330  CONTINUE                                                           PLOTC262
 340  DO 350 I=1,N                                                       PLOTC263
      IF (Y(I).GE.CUTOFF) GO TO 350                                      PLOTC264
      IF (X(I).GE.CUTOFF) GO TO 350                                      PLOTC265
      IF (CHAR(I).GE.CUTOFF) GO TO 350                                   PLOTC266
      IF (Y(I).LT.YMIN) YMIN=Y(I)                                        PLOTC267
      IF (Y(I).GT.YMAX) YMAX=Y(I)                                        PLOTC268
 350  CONTINUE                                                           PLOTC269
      DO 360 I=1,9                                                       PLOTC270
      AIM1=I-1                                                           PLOTC271
      YLABLE(I)=YMAX-(AIM1/8.0)*(YMAX-YMIN)                              PLOTC272
 360  CONTINUE                                                           PLOTC273
C                                                                        PLOTC274
C     DETERMINE THE VALUES TO BE LISTED ON THE BOTTOM HORIZONTAL AXIS    PLOTC275
C     DETERMINE XMIN, XMAX, XMID, X25 (=THE 25% POINT), AND              PLOTC276
C     X75 (=THE 75% POINT)                                               PLOTC277
C                                                                        PLOTC278
      DO 370 I=1,N                                                       PLOTC279
      IF (Y(I).GE.CUTOFF) GO TO 370                                      PLOTC280
      IF (X(I).GE.CUTOFF) GO TO 370                                      PLOTC281
      IF (CHAR(I).GE.CUTOFF) GO TO 370                                   PLOTC282
      XMIN=X(I)                                                          PLOTC283
      XMAX=X(I)                                                          PLOTC284
      GO TO 380                                                          PLOTC285
 370  CONTINUE                                                           PLOTC286
 380  DO 390 I=1,N                                                       PLOTC287
      IF (Y(I).GE.CUTOFF) GO TO 390                                      PLOTC288
      IF (X(I).GE.CUTOFF) GO TO 390                                      PLOTC289
```

```
      IF (CHAR(I).GE.CUTOFF) GO TO 390                                  PLOTC290
      IF (X(I).LT.XMIN) XMIN=X(I)                                       PLOTC291
      IF (X(I).GT.XMAX) XMAX=X(I)                                       PLOTC292
  390 CONTINUE                                                          PLOTC293
      XMID=(XMIN+XMAX)/2.0                                              PLOTC294
      X25=0.75*XMIN+0.25*XMAX                                           PLOTC295
      X75=0.25*XMIN+0.75*XMAX                                           PLOTC296
C                                                                       PLOTC297
C     BLANK OUT THE GRAPH                                               PLOTC298
C                                                                       PLOTC299
      DO 410 I=1,45                                                     PLOTC300
      DO 400 J=1,109                                                    PLOTC301
      IGRAPH(I,J)=BLANK                                                 PLOTC302
  400 CONTINUE                                                          PLOTC303
  410 CONTINUE                                                          PLOTC304
C                                                                       PLOTC305
C     PRODUCE THE VERTICAL AXES                                         PLOTC306
C                                                                       PLOTC307
      DO 420 I=3,43                                                     PLOTC308
      IGRAPH(I,5)=ALPHAI                                                PLOTC309
      IGRAPH(I,109)=ALPHAI                                              PLOTC310
  420 CONTINUE                                                          PLOTC311
      DO 430 I=3,43,5                                                   PLOTC312
      IGRAPH(I,5)=HYPHEN                                                PLOTC313
      IGRAPH(I,109)=HYPHEN                                              PLOTC314
  430 CONTINUE                                                          PLOTC315
      IGRAPH(3,1)=EQUAL                                                 PLOTC316
      IGRAPH(3,2)=ALPHAM                                                PLOTC317
      IGRAPH(3,3)=ALPHAA                                                PLOTC318
      IGRAPH(3,4)=ALPHAX                                                PLOTC319
      IGRAPH(23,1)=EQUAL                                                PLOTC320
      IGRAPH(23,2)=ALPHAM                                               PLOTC321
      IGRAPH(23,3)=ALPHAI                                               PLOTC322
      IGRAPH(23,4)=ALPHAD                                               PLOTC323
      IGRAPH(43,1)=EQUAL                                                PLOTC324
      IGRAPH(43,2)=ALPHAM                                               PLOTC325
      IGRAPH(43,3)=ALPHAI                                               PLOTC326
      IGRAPH(43,4)=ALPHAN                                               PLOTC327
C                                                                       PLOTC328
C     PRODUCE THE HORIZONTAL AXES                                       PLOTC329
C                                                                       PLOTC330
      DO 440 J=7,107                                                    PLOTC331
      IGRAPH(1,J)=HYPHEN                                                PLOTC332
      IGRAPH(45,J)=HYPHEN                                               PLOTC333
  440 CONTINUE                                                          PLOTC334
      DO 450 J=7,107,25                                                 PLOTC335
      IGRAPH(1,J)=ALPHAI                                                PLOTC336
      IGRAPH(45,J)=ALPHAI                                               PLOTC337
  450 CONTINUE                                                          PLOTC338
      DO 460 J=20,107,25                                                PLOTC339
      IGRAPH(1,J)=ALPHAI                                                PLOTC340
      IGRAPH(45,J)=ALPHAI                                               PLOTC341
  460 CONTINUE                                                          PLOTC342
C                                                                       PLOTC343
C     DETERMINE THE (X,Y) PLOT POSITIONS                                PLOTC344
C                                                                       PLOTC345
      RATIOY=40.0/(YMAX-YMIN)                                           PLOTC346
      RATIOX=100.0/(XMAX-XMIN)                                          PLOTC347
```

```
      DO 470 I=1,N                                                      PLOTC348
      IF (Y(I).GE.CUTOFF) GO TO 470                                     PLOTC349
      IF (X(I).GE.CUTOFF) GO TO 470                                     PLOTC350
      IF (CHAR(I).GE.CUTOFF) GO TO 470                                  PLOTC351
      MX=RATIOX*(X(I)-XMIN)+0.5                                         PLOTC352
      MX=MX+7                                                           PLOTC353
      MY=RATIOY*(Y(I)-YMIN)+0.5                                         PLOTC354
      MY=43-MY                                                          PLOTC355
      IARG=37                                                           PLOTC356
      IF (0.5.LT.CHAR(I).AND.CHAR(I).LT.36.5) IARG=CHAR(I)+0.5          PLOTC357
      IGRAPH(MY,MX)=IPLOTC(IARG)                                        PLOTC358
  470 CONTINUE                                                          PLOTC359
C                                                                       PLOTC360
C     WRITE OUT THE GRAPH                                               PLOTC361
C                                                                       PLOTC362
      DO 480 I=1,45                                                     PLOTC363
      IP2=I+2                                                           PLOTC364
      IFLAG=IP2-(IP2/5)*5                                               PLOTC365
      K=IP2/5                                                           PLOTC366
      IF (IFLAG.NE.0) WRITE (IPR,490) (IGRAPH(I,J),J=1,109)             PLOTC367
      IF (IFLAG.EQ.0) WRITE (IPR,500) YLABLE(K),(IGRAPH(I,J),J=1,109)   PLOTC368
  480 CONTINUE                                                          PLOTC369
      WRITE (IPR,510) XMIN,X25,XMID,X75,XMAX                            PLOTC370
      WRITE (IPR,540)                                                   PLOTC371
C                                                                       PLOTC372
  490 FORMAT (1H ,20X,109A1)                                            PLOTC373
  500 FORMAT (1H ,F20.7,109A1)                                          PLOTC374
  510 FORMAT (1H ,14X,F20.7,5X,F20.7,5X,F20.7,5X,F20.7,1X,F20.7)        PLOTC375
  520 FORMAT (1H1,59X,34HRESIDUALS VS. INDEPENDENT VARIABLE//)          PLOTC376
  530 FORMAT (1H1,53X,47HSTANDARDIZED RESIDUALS VS. INDEPENDENT VARIABLEPLOTC377
     2//)                                                               PLOTC378
  540 FORMAT (//55X,44HKNOT LOCATIONS ARE INDICATED BY THE SYMBOL X/)   PLOTC379
C                                                                       PLOTC380
      RETURN                                                            PLOTC381
      END                                                               PLOTC382
```

```
CPR*NS(1).PLOTSR(1)
      SUBROUTINE PLOTSR (X,N,NX,HOR,RES,CHAR,NKX,T,K,KX)               PLOTSR01
C-----------------------------------------------------------------------PLOTSR02
C     PLOTSR    WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING    PLOTSR03
C               DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. PLOTSR04
C               AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION PLOTSR05
C     FOR: PLOTTING KNOT LOCATIONS AND RESIDUALS/STANDARDIZED RESIDUALS PLOTSR06
C          VS. INDEPENDENT VARIABLE                                     PLOTSR07
C     SUBPROGRAMS CALLED: PLOTC                                         PLOTSR08
C     CURRENT VERSION COMPLETED MARCH 14, 1980                          PLOTSR09
C-----------------------------------------------------------------------PLOTSR10
      DIMENSION X(NX),HOR(NKX),RES(NKX),CHAR(NKX),T(KX)                 PLOTSR11
C--- CREATE SUB-VECTORS OF INDEPENDENT VARIABLE AND STANDARDIZED         PLOTSR12
C--- RESIDUALS                                                          PLOTSR13
      DO 10 I=1,N                                                       PLOTSR14
C--- SWITCH INDEPENDENT VARIABLE AND STANDARD DEVIATIONS OF RESIDUALS   PLOTSR15
      Q=HOR(I)                                                          PLOTSR16
      HOR(I)=X(I)                                                       PLOTSR17
      X(I)=Q                                                            PLOTSR18
      CHAR(I)=10.0                                                      PLOTSR19
   10 CONTINUE                                                          PLOTSR20
C--- ADD KNOT LOCATIONS TO SUB-VECTORS                                  PLOTSR21
      DO 20 I=1,K                                                       PLOTSR22
      L=I+N                                                             PLOTSR23
      HOR(L)=T(I)                                                       PLOTSR24
      RES(L)=0.0                                                        PLOTSR25
      CHAR(L)=0.0                                                       PLOTSR26
   20 CONTINUE                                                          PLOTSR27
C--- GENERATE PLOT OF RESIDUALS VS. INDEPENDENT VARIABLE                PLOTSR28
      NK=N+K                                                            PLOTSR29
      CALL PLOTC (RES,HOR,CHAR,NK,1)                                    PLOTSR30
C--- CREATE SUB-VECTOR OF STANDARDIZED RESIDUALS                        PLOTSR31
      DO 30 I=1,N                                                       PLOTSR32
      IF (X(I).LE.0.0) GO TO 30                                         PLOTSR33
      RES(I)=RES(I)/X(I)                                                PLOTSR34
   30 CONTINUE                                                          PLOTSR35
C--- GENERATE PLOT OF STANDARDIZED RESIDUALS VS. INDEPENDENT VARIABLE   PLOTSR36
      CALL PLOTC (RES,HOR,CHAR,NK,2)                                    PLOTSR37
      RETURN                                                            PLOTSR38
      END                                                               PLOTSR39
```

```
CPR*NS(1).PPREP(2)
         SUBROUTINE PPREP (T,BCOEF,SCRTCH,BREAK,COEF,KX,JX,NB,MO,IP)         PPREP001
C-----------------------------------------------------------------------PPREP002
C    PPREP   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING          PPREP003
C            DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.      PPREP004
C            AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION      PPREP005
C    FOR: CONVERTING THE B-REPRESENTATION OF THE SPLINE INTO THE           PPREP006
C         PIECEWISE POLYNOMIAL REPRESENTATION                              PPREP007
C    SUBPROGRAMS CALLED: BSPLPP                                            PPREP008
C    CURRENT VERSION COMPLETED APRIL 3, 1980                               PPREP009
C-----------------------------------------------------------------------PPREP010
      DIMENSION T(KX),BCOEF(KX),SCRTCH(JX,JX),BREAK(KX),COEF(JX,KX),    PPREP011
     2 II(20)                                                           PPREP012
   10 FORMAT (//1X,50(1H-)/1X,38H* PIECEWISE POLYNOMIAL REPRESENTATION ,PPREP013
     2 12HOF SPLINES */1X,50(1H-)//9X,18H.....INTERVAL.....,9X,         PPREP014
     3 27HCOEFFICIENTS OF (X-X(I))**P//3X,1HI,6X,4HX(I),7X,6HX(I+1),5X, PPREP015
     4 3HP =,8(I4,8X)/35X,8(I4,8X)/35X,4(I4,8X))                        PPREP016
   20 FORMAT (1X,I3,2X,2G12.5,3X,8G12.5/33X,8G12.5/33X,4G12.5)          PPREP017
   30 FORMAT ()                                                         PPREP018
   40 FORMAT (/1X,42H******* PRINTOUT OF PIECEWISE POLYNOMIALS ,        PPREP019
     2 18HSUPPRESSED *******)                                           PPREP020
C--- COMPUTE PIECEWISE POLYNOMIAL REPRESENTATION OF SPLINE              PPREP021
      CALL BSPLPP (T,BCOEF,NB,MO,SCRTCH,BREAK,COEF,L,JX)                PPREP022
C--- DIVIDE EACH COEF(J,I) BY (J-1) FACTORIAL TO NORMALIZE              PPREP023
      IF (MO.LT.3) GO TO 80                                             PPREP024
      DO 70 I=1,L                                                       PPREP025
      DO 60 J=3,MO                                                      PPREP026
      DO 50 K=3,J                                                       PPREP027
      COEF(J,I)=COEF(J,I)/FLOAT(K-1)                                    PPREP028
   50 CONTINUE                                                          PPREP029
   60 CONTINUE                                                          PPREP030
   70 CONTINUE                                                          PPREP031
   80 IF (IP.EQ.0) GO TO 110                                            PPREP032
      DO 90 I=1,MO                                                      PPREP033
      II(I)=I-1                                                         PPREP034
   90 CONTINUE                                                          PPREP035
      WRITE (6,10) (II(I),I=1,MO)                                       PPREP036
      WRITE (6,30)                                                      PPREP037
      DO 100 I=1,L                                                      PPREP038
      WRITE (6,20) I,BREAK(I),BREAK(I+1),(COEF(J,I),J=1,MO)             PPREP039
  100 CONTINUE                                                          PPREP040
      RETURN                                                            PPREP041
  110 WRITE (6,40)                                                      PPREP042
      RETURN                                                            PPREP043
      END                                                               PPREP044
```

CPR*NS(1).RESSD(1)

```
      SUBROUTINE RESSD (X,Y,W,N,NX,NKX,NRSD,T,BCOEF,XXI,K,KX,NB,MO,YHAT,RESSD001
     2 RES,RSD,BIATX,JX,IP)                                            RESSD002
C-----------------------------------------------------------------------RESSD003
C     RESSD   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING      RESSD004
C             DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.  RESSD005
C             AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION  RESSD006
C     FOR: COMPUTING PREDICTED Y-VALUES, STANDARD DEVIATIONS OF         RESSD007
C          PREDICTED Y-VALUES, AND THE RESIDUAL STANDARD DEVIATION      RESSD008
C     SUBPROGRAMS CALLED: BVALUE, INTERV, BSPLVB                        RESSD009
C     CURRENT VERSION COMPLETED MARCH 24, 1980                          RESSD010
C-----------------------------------------------------------------------RESSD011
      DIMENSION X(NX),Y(NX),W(NX),T(NX),BCOEF(KX),YHAT(NKX),RES(NKX),   RESSD012
     2 BIATX(JX),XXI(KX,KX)                                             RESSD013
10    FORMAT (//1X,25(1H-)/1X,25H* ANALYSIS OF RESIDUALS */1X,25(1H-)// RESSD014
     2 9X,6HWEIGHT,20X,8HOBSERVED,5X,9HPREDICTED,20X,10HSTD DEV OF/4X,  RESSD015
     3 1HI,5X,4HW(I),9X,4HX(I),10X,4HY(I),10X,4HY(I),6X,11HRESIDUAL(I), RESSD016
     4 3X,14HPREDICTED Y(I)/)                                          RESSD017
20    FORMAT (1X,I4,2X,G11.5,3G14.7,G12.5,G16.7)                       RESSD018
30    FORMAT (//5X,16HRESIDUAL STD DEV,5X,13HRESIDUAL D.F./7X,G12.6,9X, RESSD019
     2 I5)                                                              RESSD020
40    FORMAT (/1X,48H******* PRINTOUT OF RESIDUALS SUPPRESSED *******)  RESSD021
C--- INITIALIZE SUMMING VARIABLE                                        RESSD022
      SUM=0.0                                                           RESSD023
      IF (IP.EQ.0) WRITE (6,40)                                         RESSD024
      IF (IP.NE.0) WRITE (6,10)                                         RESSD025
C--- COMPUTE PREDICTED VALUES AND RESIDUALS                             RESSD026
      DO 50 I=1,N                                                       RESSD027
      XX=X(I)                                                           RESSD028
      YHAT(I)=BVALUE(T,BCOEF,NB,MO,XX,0)                                RESSD029
      RES(I)=Y(I)-YHAT(I)                                               RESSD030
      SUM=SUM+W(I)*RES(I)**2                                            RESSD031
50    CONTINUE                                                          RESSD032
C--- COMPUTE RESIDUAL STANDARD DEVIATION                                RESSD033
      RSD=SQRT(SUM/FLOAT(NRSD))                                         RESSD034
C--- COMPUTE STANDARD DEVIATIONS OF PREDICTED VALUES                    RESSD035
      DO 90 L=1,N                                                       RESSD036
      XX=X(L)                                                           RESSD037
C--- FIND INDEX OF FIRST KNOT TO LEFT OF X-VALUE                        RESSD038
      CALL INTERV (T,K,XX,LEFT,MFLAG)                                   RESSD039
C--- CHECK WHETHER X-VALUE LISE WITHIN KNOT SPAN                        RESSD040
      IF (MFLAG.EQ.0) GO TO 60                                          RESSD041
C--- SET RESIDUAL TO ZERO FOR X-VALUE OUTSIDE KNOT SPAN                 RESSD042
      RES(L)=0.0                                                        RESSD043
C--- SET STANDARD DEVIATION OF RESIDUAL TO ZERO                         RESSD044
      YHAT(L)=0.0                                                       RESSD045
      CO TO 90                                                          RESSD046
C--- EVALUATE POSSIBLY NON-ZERO B-SPLINES AT X-VALUE                    RESSD047
60    CALL BSPLVB (T,MO,1,XX,LEFT,BIATX)                                RESSD048
C--- COMPUTE VARIANCE COEFFICIENT (BIATX)'(XXI)(BIATX) OF               RESSD049
C--- PREDICTED Y-VALUE                                                  RESSD050
      NLOW=LEFT-MO                                                      RESSD051
      Q1=0.0                                                            RESSD052
      DO 80 I=1,MO                                                      RESSD053
      Q2=0.0                                                            RESSD054
      NI=NLOW+I                                                         RESSD055
      DO 70 J=1,MO                                                      RESSD056
      NJ=NLOW+J                                                         RESSD057
```

```
          Q2=Q2+BIATX(J)*XXI(NJ,NI)                                    RESSD058
   70     CONTINUE                                                     RESSD059
          Q1=Q1+Q2*BIATX(I)                                            RESSD060
   80     CONTINUE                                                     RESSD061
C---  COMPUTE STANDARD DEVIATION OF PREDICTED Y-VALUE                  RESSD062
          YHATSD=RSD*SQRT(Q1)                                          RESSD063
          IF (IP.EQ.0) GO TO 90                                        RESSD064
          WRITE (6,20) L,W(L),X(L),Y(L),YHAT(L),RES(L),YHATSD          RESSD065
C---  COMPUTE STANDARD DEVIATION OF EACH RESIDUAL AND STORE IN         RESSD066
C---  VECTOR *YHAT*                                                    RESSD067
          YHAT(L)=SQRT(RSD**2-YHATSD**2)                               RESSD068
   90     CONTINUE                                                     RESSD069
          WRITE (6,30) RSD,NRSD                                        RESSD070
          RETURN                                                       RESSD071
          END                                                          RESSD072
```

CPR*NS(1).RSQ(3)

```
      SUBROUTINE RSQ (RSD,NRSD,Y,W,N,NX,NNZ)                            RSQ00001
C-----------------------------------------------------------------------RSQ00002
C     RSQ   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING        RSQ00003
C           DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.    RSQ00004
C           AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION    RSQ00005
C     FOR: COMPUTING THE MULTIPLE CORRELATION COEFFICIENT R SQUARE      RSQ00006
C     SUBPROGRAMS CALLED: -NONE-                                        RSQ00007
C     CURRENT VERSION COMPLETED JUNE 11, 1980                           RSQ00008
C-----------------------------------------------------------------------RSQ00009
      DIMENSION Y(NX),W(NX)                                             RSQ00010
10    FORMAT (//7X,8HR SQUARE,7X,26HNUMBER OF NON-ZERO WEIGHTS/4X,F11.8,RSQ00011
     2 16X,I5)                                                          RSQ00012
C--- COMPUTE RESIDUAL SUM OF SQUARES                                    RSQ00013
      RSS=FLOAT(NRSD)*RSD**2                                            RSQ00014
C--- INITIALIZE SUMMING VARIABLE                                        RSQ00015
      TSS=0.0                                                           RSQ00016
      DO 20 I=1,N                                                       RSQ00017
      TSS=TSS+Y(I)*SQRT(W(I))                                           RSQ00018
20    CONTINUE                                                          RSQ00019
      YM=TSS/FLOAT(NNZ)                                                 RSQ00020
C--- INITIALIZE SUMMING VARIABLE                                        RSQ00021
      TSS=0.0                                                           RSQ00022
C--- COMPUTE TOTAL SUM OF SQUARES                                       RSQ00023
      DO 30 I=1,N                                                       RSQ00024
      IF (W(I).EQ.0.0) GO TO 30                                         RSQ00025
      TSS=TSS+(Y(I)*SQRT(W(I))-YM)**2                                   RSQ00026
30    CONTINUE                                                          RSQ00027
C--- COMPUTE R**2                                                       RSQ00028
      R2=1.0-RSS/TSS                                                    RSQ00029
C--- TO PRINT OUT R-SQUARED CHANGE THE 'C' IN THE FOLLOWING LINE        RSQ00030
C--- TO A BLANK.                                                        RSQ00031
C     WRITE (6,10) R2,NNZ                                               RSQ00032
      RETURN                                                            RSQ00033
      END                                                               RSQ00034
```

CPR*NS(1).SDYFIN(1)

```
      SUBROUTINE SDYFIN (XF,YFSD,NF,T,K,MO,XXI,KX,RSD,BIATX,JX)         SDYFIN01
C-----------------------------------------------------------------------SDYFIN02
C     SDYFIN   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING     SDYFIN03
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. SDYFIN04
C              AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION SDYFIN05
C     FOR: COMPUTING THE STANDARD DEVIATION OF THE PREDICTED            SDYFIN06
C          Y-VALUES IN THE FINE MESH                                    SDYFIN07
C     SUBPROGRAMS CALLED: INTERV, BSPLVB                                SDYFIN08
C     CURRENT VERSION COMPLETED OCTOBER 9, 1979                         SDYFIN09
C-----------------------------------------------------------------------SDYFIN10
      DIMENSION XF(NF),YFSD(NF),T(KX),BIATX(JX),XXI(KX,KX)              SDYFIN11
10    FORMAT (/5X,18H<<<<< STD. DEV. OF,I5,1X,11HPREDICTED Y,1X,        SDYFIN12
     2 21HVALUES COMPUTED >>>>>)                                        SDYFIN13
      DO 40 L=1,NF                                                      SDYFIN14
      XX=XF(L)                                                          SDYFIN15
C--- FIND INDEX OF FIRST KNOT TO LEFT OF X-VALUE                        SDYFIN16
      CALL INTERV (T,K,XX,LEFT,MFLAG)                                   SDYFIN17
C--- EVALUATE POSSIBLY NON-ZERO B-SPLINES AT X-VALUE                    SDYFIN18
      CALL BSPLVB (T,MO,1,XX,LEFT,BIATX)                                SDYFIN19
C--- COMPUTE VARIANCE COEFFICIENT (BIATX)'(XXI)(BIATX) OF               SDYFIN20
C--- PREDICTED Y-VALUE                                                  SDYFIN21
      NLOW=LEFT-MO                                                      SDYFIN22
      Q1=0.0                                                            SDYFIN23
      DO 30 I=1,MO                                                      SDYFIN24
      Q2=0.0                                                            SDYFIN25
      NI=NLOW+I                                                         SDYFIN26
      DO 20 J=1,MO                                                      SDYFIN27
      NJ=NLOW+J                                                         SDYFIN28
      Q2=Q2+BIATX(J)*XXI(NJ,NI)                                         SDYFIN29
20    CONTINUE                                                          SDYFIN30
      Q1=Q1+Q2*BIATX(I)                                                 SDYFIN31
30    CONTINUE                                                          SDYFIN32
C--- COMPUTE STANDARD DEVIATION OF PREDICTED Y-VALUE                    SDYFIN33
      YFSD(L)=RSD*SQRT(Q1)                                              SDYFIN34
40    CONTINUE                                                          SDYFIN35
      WRITE (6,10) NF                                                   SDYFIN36
      RETURN                                                            SDYFIN37
      END                                                               SDYFIN38
```

```
CPR*NS(1).SORT1(2)
      SUBROUTINE SORT1 (X,M,N,NX)                                       SORT1001
C-----------------------------------------------------------------------SORT1002
C   SORT1   OBTAINED BY CHARLES P. REEVE FROM DR. D. A. ZAHN AT         SORT1003
C           THE FLORIDA STATE UNIVERSITY, TALLAHASSEE, FLORIDA          SORT1004
C           UNDER THE NAME *FTASORT*.  SEVERAL MINOR CORRECTIONS        SORT1005
C           WERE MADE TO THE ORIGINAL VERSION AFTER LINE 61.            SORT1006
C   FOR: SORTING THE SEGMENT OF A REAL ARRAY BETWEEN ENTRIES            SORT1007
C        M AND N FROM SMALLEST TO LARGEST                               SORT1008
C   SUBROUTINES CALLED: -NONE-                                          SORT1009
C   CURRENT VERSION COMPLETED JANUARY 30, 1978                          SORT1010
C   NOTE: ARRAYS LP(K) AND LQ(K) PERMIT SORTING UP TO 2**(K+1)-1        SORT1011
C         ELEMENTS, I.E., FOR K=25 YOU MAY SORT 67,108,863 ELEMENTS.    SORT1012
C-----------------------------------------------------------------------SORT1013
      DIMENSION LP(25),LQ(25),X(NX)                                     SORT1014
      I=1                                                               SORT1015
      KM=M                                                              SORT1016
      KN=N                                                              SORT1017
      IF (KM.GE.KN) GO TO 70                                            SORT1018
   10 J=KM                                                              SORT1019
      K=(KN+KM)/2                                                       SORT1020
      A=X(K)                                                            SORT1021
      IF (X(KM).LE.A) GO TO 20                                          SORT1022
      X(K)=X(KM)                                                        SORT1023
      X(KM)=A                                                           SORT1024
      A=X(K)                                                            SORT1025
   20 L=KN                                                              SORT1026
      IF (X(KN).GE.A) GO TO 40                                          SORT1027
      X(K)=X(KN)                                                        SORT1028
      X(KN)=A                                                           SORT1029
      A=X(K)                                                            SORT1030
      IF (X(KM).LE.A) GO TO 40                                          SORT1031
      X(K)=X(KM)                                                        SORT1032
      X(KM)=A                                                           SORT1033
      A=X(K)                                                            SORT1034
      GO TO 40                                                          SORT1035
   30 X(L)=X(J)                                                         SORT1036
      X(J)=B                                                            SORT1037
   40 L=L-1                                                             SORT1038
      IF (X(L).GT.A) GO TO 40                                           SORT1039
      B=X(L)                                                            SORT1040
   50 J=J+1                                                             SORT1041
      IF (X(J).LT.A) GO TO 50                                           SORT1042
      IF (J.LE.L) GO TO 30                                              SORT1043
      IF (L-KM.LE.KN-J) GO TO 60                                        SORT1044
      LP(I)=KM                                                          SORT1045
      LQ(I)=L                                                           SORT1046
      KM=J                                                              SORT1047
      I=I+1                                                             SORT1048
      GO TO 80                                                          SORT1049
   60 LP(I)=J                                                           SORT1050
      LQ(I)=KN                                                          SORT1051
      KN=L                                                              SORT1052
      I=I+1                                                             SORT1053
      GO TO 80                                                          SORT1054
   70 I=I-1                                                             SORT1055
      IF (I.EQ.0) RETURN                                                SORT1056
      KM=LP(I)                                                          SORT1057
```

```
      KN=LQ(I)                                                    SORT1058
   80 IF (KN-KM.GE.11) GO TO 10                                   SORT1059
      KM=KM-1                                                     SORT1060
   90 KM=KM+1                                                     SORT1061
      IF (KM.EQ.KN) GO TO 70                                      SORT1062
      A=X(KM+1)                                                   SORT1063
      IF (X(KM).LE.A) GO TO 90                                    SORT1064
      J=KM                                                        SORT1065
  100 X(J+1)=X(J)                                                 SORT1066
      J=J-1                                                       SORT1067
      IF (J.EQ.M-1) GO TO 110                                     SORT1068
      IF (A.LT.X(J)) GO TO 100                                    SORT1069
  110 X(J+1)=A                                                    SORT1070
      GO TO 90                                                    SORT1071
      END                                                         SORT1072
```

```
CPR*NS(1).SORT2(3)
1          SUBROUTINE SORT2 (X,Y,M,N,NX)                                       SORT2001
2     C-------------------------------------------------------------------------SORT2002
3     C    SORT2   OBTAINED BY CHARLES P. REEVE FROM DR. D. A. ZAHN AT          SORT2003
4     C            THE FLORIDA STATE UNIVERSITY, TALLAHASSEE, FLORIDA           SORT2004
5     C            UNDER THE NAME *FTASORT*.  SEVERAL MINOR CORRECTIONS         SORT2005
6     C            WERE MADE TO THE ORIGINAL VERSION AFTER LINE 61.             SORT2006
7     C            THE PROGRAM HAS BEEN ALTERED SO THAT A SECOND ARRAY IS       SORT2007
8     C            CARRIED ALONG AND SORTED IDENTICALLY AS THE FIRST            SORT2008
9     C    FOR: SORTING THE SEGMENT OF A REAL ARRAY BETWEEN ENTRIES             SORT2009
10    C         M AND N FROM SMALLEST TO LARGEST                                SORT2010
11    C    SUBROUTINES CALLED: -NONE-                                           SORT2011
12    C    CURRENT VERSION COMPLETED MARCH 24, 1980                             SORT2012
13    C    NOTE: ARRAYS LP(K) AND LQ(K) PERMIT SORTING UP TO 2**(K+1)-1         SORT2013
14    C          ELEMENTS, I.E., FOR K=25 YOU MAY SORT 67,108,863 ELEMENTS.     SORT2014
15    C-------------------------------------------------------------------------SORT2015
16         DIMENSION LP(25),LQ(25),X(NX),Y(NX)                                 SORT2016
17         I=1                                                                 SORT2017
18         KM=M                                                                SORT2018
19         KN=N                                                                SORT2019
20         IF (KM.GE.KN) GO TO 70                                              SORT2020
21    10   J=KM                                                                SORT2021
22         K=(KN+KM)/2                                                         SORT2022
23         A=X(K)                                                              SORT2023
24         B=Y(K)                                                              SORT2024
25         IF (X(KM).LE.A) GO TO 20                                            SORT2025
26         X(K)=X(KM)                                                          SORT2026
27         Y(K)=Y(KM)                                                          SORT2027
28         X(KM)=A                                                             SORT2028
29         Y(KM)=B                                                             SORT2029
30         A=X(K)                                                              SORT2030
31         B=Y(K)                                                              SORT2031
32    20   L=KN                                                                SORT2032
33         IF (X(KN).GE.A) GO TO 40                                            SORT2033
34         X(K)=X(KN)                                                          SORT2034
35         Y(K)=Y(KN)                                                          SORT2035
36         X(KN)=A                                                             SORT2036
37         Y(KN)=B                                                             SORT2037
38         A=X(K)                                                              SORT2038
39         B=Y(K)                                                              SORT2039
40         IF (X(KM).LE.A) GO TO 40                                            SORT2040
41         X(K)=X(KM)                                                          SORT2041
42         Y(K)=Y(KM)                                                          SORT2042
43         X(KM)=A                                                             SORT2043
44         Y(KM)=B                                                             SORT2044
45         A=X(K)                                                              SORT2045
46         B=Y(K)                                                              SORT2046
47         GO TO 40                                                            SORT2047
48    30   X(L)=X(J)                                                           SORT2048
49         Y(L)=Y(J)                                                           SORT2049
50         X(J)=G                                                              SORT2050
51         Y(J)=H                                                              SORT2051
52    40   L=L-1                                                               SORT2052
53         IF (X(L).GT.A) GO TO 40                                             SORT2053
54         G=X(L)                                                              SORT2054
55         H=Y(L)                                                              SORT2055
56    50   J=J+1                                                               SORT2056
57         IF (X(J).LT.A) GO TO 50                                             SORT2057
```

```
      IF (J.LE.L) GO TO 30                                        SORT2058
      IF (L-KM.LE.KN-J) GO TO 60                                  SORT2059
      LP(I)=KM                                                    SORT2060
      LQ(I)=L                                                     SORT2061
      KM=J                                                        SORT2062
      I=I+1                                                       SORT2063
      GO TO 80                                                    SORT2064
60    LP(I)=J                                                     SORT2065
      LQ(I)=KN                                                    SORT2066
      KN=L                                                        SORT2067
      I=I+1                                                       SORT2068
      GO TO 80                                                    SORT2069
70    I=I-1                                                       SORT2070
      IF (I.EQ.0) RETURN                                          SORT2071
      KM=LP(I)                                                    SORT2072
      KN=LQ(I)                                                    SORT2073
80    IF (KN-KM.GE.11) GO TO 10                                   SORT2074
      KM=KM-1                                                     SORT2075
90    KM=KM+1                                                     SORT2076
      IF (KM.EQ.KN) GO TO 70                                      SORT2077
      A=X(KM+1)                                                   SORT2078
      B=Y(KM+1)                                                   SORT2079
      IF (X(KM).LE.A) GO TO 90                                    SORT2080
      J=KM                                                        SORT2081
100   X(J+1)=X(J)                                                 SORT2082
      Y(J+1)=Y(J)                                                 SORT2083
      J=J-1                                                       SORT2084
      IF (J.EQ.M-1) GO TO 110                                     SORT2085
      IF (A.LT.X(J)) GO TO 100                                    SORT2086
110   X(J+1)=A                                                    SORT2087
      Y(J+1)=B                                                    SORT2088
      GO TO 90                                                    SORT2089
      END                                                         SORT2090
```

CPR*NS(1).SPLEEN(8)

```
      SUBROUTINE SPLEEN (H,X,Y,W,R1,R2,RES,N,NX,NKX,T,BCOEF,XXI,Q,DIAG,KSPLEE001
     2,KX,YY,NY,NYX,MD,SCRTCH,JX,AL,DL,C,IP)                             SPLEE002
C-----------------------------------------------------------------------SPLEE003
C     SPLEEN   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING      SPLEE004
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.  SPLEE005
C                                                                        SPLEE006
C           * * * * * * * * * * * * * * * * * * * * * * *                SPLEE007
C           * FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION *                SPLEE008
C           * * * * * * * * * * * * * * * * * * * * * * *                SPLEE009
C                                                                        SPLEE010
C     THIS PACKAGE OF SUBROUTINES WAS WRITTEN FOR THE FOLLOWING          SPLEE011
C     CALIBRATION PROCEDURES:                                            SPLEE012
C                                                                        SPLEE013
C     1) A MONOTONIC SEQUENCE OF RESPONSES Y(1), Y(2), ... , Y(N)        SPLEE014
C        EACH CONTAINING SOME ERROR ARE OBSERVED AT KNOWN POINTS         SPLEE015
C        X(1), X(2), ... , X(N) WHERE X(1) < X(2) < ... < X(N).          SPLEE016
C                                                                        SPLEE017
C     2) A SPLINE OF SPECIFIED DEGREE WITH A SPECIFIED SEQUENCE OF       SPLEE018
C        FIXED KNOTS IS FIT TO THE Y-VALUES WHICH MAY BE WEIGHTED.       SPLEE019
C                                                                        SPLEE020
C     3) THE RESIDUAL STANDARD DEVIATION IS COMPUTED IN ORDER TO         SPLEE021
C        MEASURE THE GOODNESS OF THE SPLINE FIT.                         SPLEE022
C                                                                        SPLEE023
C     4) PREDICTED RESPONSE VALUES ARE COMPUTED AT A LARGE NUMBER        SPLEE024
C        OF UNIFORMLY SPACED X-VALUES BETWEEN THE EXTREME KNOTS.         SPLEE025
C        A CONFIDENCE INTERVAL FOR EACH PREDICTED RESPONSE IS            SPLEE026
C        COMPUTED BASED ON SPECIFIED CONSTANTS ALPHA, BETA, AND C        SPLEE027
C        IN ACCORDANCE WITH REFERENCE PAPER BY SCHEFFE GIVEN BELOW.      SPLEE028
C                                                                        SPLEE029
C     5) FOR SPECIFIED Y-VALUES, INVERSE INTERPOLATION IS APPLIED        SPLEE030
C        TO THE CALIBRATION CURVE AND ITS CONFIDENCE BAND TO GIVE        SPLEE031
C        PREDICTED X-VALUES WITH CORRESPONDING UPPER AND LOWER           SPLEE032
C        CONFIDENCE LIMITS.                                              SPLEE033
C                                                                        SPLEE034
C     PASSED PARAMETERS (AND DIMENSIONS):                                SPLEE035
C                                                                        SPLEE036
C        * H(80) = UP TO 80 CHARACTERS IN 80A1 FORMAT IDENTIFYING THE    SPLEE037
C                  DATA                                                  SPLEE038
C                                                                        SPLEE039
C        * X(NX) = VECTOR (LENGTH N) OF X-VALUES WHERE OBSERVATIONS      SPLEE040
C                  WERE MADE (INDEPENDENT VARIABLE)                      SPLEE041
C                                                                        SPLEE042
C        * Y(NX) = VECTOR (LENGTH N) OF OBSERVATIONS                     SPLEE043
C                                                                        SPLEE044
C        * W(NX) = VECTOR (LENGTH N) OF WEIGHTS FOR OBSERVATIONS         SPLEE045
C                                                                        SPLEE046
C        R1(NKX) = VECTOR (LENGTH N+K) FOR SCRATCH AREA                  SPLEE047
C                                                                        SPLEE048
C        R2(NKX) = VECTOR (LENGTH N+K) FOR SCRATCH AREA                  SPLEE049
C                                                                        SPLEE050
C       RES(NKX) = VECTOR (LENGTH N+K) OF RESIDUALS FROM SPLINE FIT      SPLEE051
C                                                                        SPLEE052
C            * N = NUMBER OF OBSERVATIONS                                SPLEE053
C                                                                        SPLEE054
C           * NX = DIMENSION (>=N) OF VECTORS X,Y,W                      SPLEE055
C                                                                        SPLEE056
C          * NKX = DIMENSION (>=N+K) OF VECTORS R1,R2,RES                SPLEE057
```

```
C                                                                      SPLEE058
C        * T(KX) = VECTOR (LENGTH K+2*MD) OF KNOT LOCATIONS            SPLEE059
C                                                                      SPLEE060
C     BCOEF(KX) = VECTOR (LENGTH K+MD-1) OF B-SPLINE COEFFICIENTS      SPLEE061
C                                                                      SPLEE062
C    XXI(KX,KX) = VARIANCE-COVARIANCE MATRIX (SIZE [K+MD-1]X[K+MD-1])  SPLEE063
C                 OF B-SPLINE COEFFICIENTS                             SPLEE064
C                                                                      SPLEE065
C      Q(JX,KX) = MATRIX (SIZE [MD+1]X[K+MD-1]) FOR SCRATCH AREA       SPLEE066
C                                                                      SPLEE067
C      DIAG(KX) = VECTOR (LENGTH K+MD-1) FOR SCRATCH AREA              SPLEE068
C                                                                      SPLEE069
C           * K = NUMBER OF KNOTS SPECIFIED BY USER (LATER INCREASED   SPLEE070
C                 TO K+2*MD BY PROGRAM)                                SPLEE071
C                                                                      SPLEE072
C          * KX = DIMENSION (>=K+2*MD) OF VECTORS T,BCOEF,DIAG AND     SPLEE073
C                 MATRICES XXI AND Q (COLUMN ONLY)                     SPLEE074
C                                                                      SPLEE075
C     * YY(NYX) = VECTOR (LENGTH NY) OF Y-VALUES FOR WHICH PREDICTED   SPLEE076
C                 X-VALUES (WITH CONFIDENCE INTERVALS) ARE TO BE       SPLEE077
C                 COMPUTED                                             SPLEE078
C                                                                      SPLEE079
C          * NY = NUMBER OF Y-VALUES FOR WHICH PRECICTED X-VALUES      SPLEE080
C                 ARE TO BE COMPUTED                                   SPLEE081
C                                                                      SPLEE082
C         * NYX = DIMENSION (>=NY) OF VECTOR YY                        SPLEE083
C                                                                      SPLEE084
C          * MD = DEGREE OF SPLINE (<=19); FOR EXAMPLE, 1=LINEAR,      SPLEE085
C                 2=QUADRATIC, 3=CUBIC)                                SPLEE086
C                                                                      SPLEE087
C SCRTCH(JX,JX) = MATRIX (SIZE [MD+1]X[MD+1]) FOR SCRATCH AREA         SPLEE088
C                                                                      SPLEE089
C          * JX = DIMENSION OF SQUARE MATRIX SCRTCH AND ROW            SPLEE090
C                 DIMENSION OF MATRIX Q = 20                           SPLEE091
C                                                                      SPLEE092
C          * AL = ALPHA LEVEL OF SIGNIFICANCE (SEE REFERENCE BELOW)    SPLEE093
C                                                                      SPLEE094
C          * DL = DELTA LEVEL OF SIGNIFICANCE (SEE REFERENCE BELOW)    SPLEE095
C                                                                      SPLEE096
C           * C = CONSTANT IN THE INTERVAL (0.85,1.25) ASSOCIATED      SPLEE097
C                 WITH SCHEFFE'S CALIBRATION TECHNIQUE                 SPLEE098
C                                                                      SPLEE099
C                 0  FOR ABBREVIATED PRINTOUT (NO RESIDUALS)           SPLEE100
C          * IP = 1  FOR FULL PRINTOUT (RESIDUALS, PP REPRESENTATION)  SPLEE101
C                 2  FOR FULL PRINTOUT PLUS Y-CONFIDENCE INTERVALS FOR SPLEE102
C                    300 EVENLY SPACED X-VALUES OVER KNOT SPAN         SPLEE103
C                                                                      SPLEE104
C   * INDICATES THAT AN INPUT VALUE IS REQUIRED FOR THIS VARIABLE      SPLEE105
C                                                                      SPLEE106
C   NOTE: THE USER IS NOT REQUIRED TO ORDER THE ELEMENTS OF ANY INPUT  SPLEE107
C         VECTOR.  THE PROGRAM WILL AUTOMATICALLY ORDER THOSE VECTORS  SPLEE108
C         WHICH NEED TO BE ORDERED.                                    SPLEE109
C                                                                      SPLEE110
C   REFERENCE: SCHEFFE, HENRY, 'A STATISTICAL THEORY OF CALIBRATION',  SPLEE111
C              THE ANNALS OF STATISTICS, VOLUME 1, NUMBER 1,           SPLEE112
C              JANUARY 1973, PP. 1-37                                  SPLEE113
C                                                                      SPLEE114
C   SUBPROGRAMS CALLED: ADKNTS, CHECK1, CHECK2, CIYFIN, COVAR, L2APPR, SPLEE115
```

```
C                        PLOTSR, PPREP, RESSD, RSQ, SDYFIN, SORT1,       SPLEE116
C                        SORT2, XYFINE, YTOXCI                            SPLEE117
C    CURRENT VERSION COMPLETED JUNE 11, 1980                              SPLEE118
C-----------------------------------------------------------------------SPLEE119
C--- SET DIMENSIONS OF VECTORS AND MATRIX                                 SPLEE120
      DIMENSION X(NX),Y(NX),W(NX),R1(NKX),R2(NKX),RES(NKX)                SPLEE121
      DIMENSION T(KX),Q(JX,KX),DIAG(KX),BCOEF(KX),XXI(KX,KX)              SPLEE122
      DIMENSION YY(NYX),SCRTCH(JX,JX),BIATX(20),H(80)                     SPLEE123
C     PARAMETER NF=300                                                    SPLEE124
      DIMENSION XF(300),YF(300),YFL(300),YFU(300),YFSD(300)               SPLEE125
10    FORMAT (1H1/1X,45(1H*)/1X,32H* FIXED-KNOT SPLINE PACKAGE FOR ,      SPLEE126
     2 13HCALIBRATION */1X,45(1H*))                                       SPLEE127
20    FORMAT (//1X,39(1H-)/1X,25H* ESTIMATION OF B-SPLINE ,               SPLEE128
     2 14HCOEFFICIENTS */1X,39(1H-))                                      SPLEE129
30    FORMAT (/9X,8HB-SPLINE/4X,1HI,5X,4HCOEF,10X,7HSTD DEV/)             SPLEE130
40    FORMAT (1X,I4,2G15.8)                                               SPLEE131
50    FORMAT (/1X,42H******* PRINTOUT OF B-SPLINE COEFFICIENTS ,          SPLEE132
     2 18HSUPPRESSED *******)                                             SPLEE133
60    FORMAT (//1X,42(1H-)/1X,37H* PARAMETERS OF LEAST SQUARES SPLINE ,  SPLEE134
     2 5HFIT */1X,42(1H-)//13X,18HDEGREE OF SPLINE =,I4//3X,              SPLEE135
     3 28HNUMBER OF     OBSERVATIONS =,I4/3X,                             SPLEE136
     4 28HNUMBER OF     ZERO WEIGHTS =,I4/3X,19HNUMBER OF NON-ZERO ,      SPLEE137
     5 9HWEIGHTS =,I4/3X,28HNUMBER OF             KNOTS =,I4/3X,          SPLEE138
     6 28HNUMBER OF        B-SPLINES =,I4//11X,18HNUMBER OF Y-VALUES/7X  SPLEE139
     7,24HFOR WHICH X CONFIDENCE =,I4/3X,18HINTERVAL IS TO BE ,           SPLEE140
     8 8HCOMPUTED)                                                        SPLEE141
70    FORMAT (//5X,25H----- FULL PRINTOUT -----/)                         SPLEE142
80    FORMAT (//5X,32H----- ABBREVIATED PRINTOUT -----/)                  SPLEE143
90    FORMAT (//1X,80A1)                                                  SPLEE144
100   FORMAT (//1X,8(1H*)/1X,8H* STOP */1X,8(1H*)/)                       SPLEE145
C--- DEFINE NUMBER OF POINTS IN FINE MESH                                 SPLEE146
      NF=300                                                              SPLEE147
C--- WRITE HEADING FOR HARDCOPY OUTPUT                                    SPLEE148
      WRITE (6,10)                                                        SPLEE149
C--- WRITE RUN IDENTIFICATION                                             SPLEE150
      WRITE (6,90) (H(I),I=1,80)                                          SPLEE151
      IF (IP.GE.1) WRITE (6,70)                                           SPLEE152
      IF (IP.EQ.0) WRITE (6,80)                                           SPLEE153
C--- COMPUTE ORDER OF SPLINE                                              SPLEE154
      MO=MD+1                                                             SPLEE155
C--- CHECK THAT INPUT PARAMETERS FALL WITHIN ALLOWABLE RANGES             SPLEE156
C                                                                         SPLEE157
      CALL CHECK1 (W,N,NX,K,KX,NKX,NY,NYX,JX,MO,AL,DL,C,NZ)               SPLEE158
C                                                                         SPLEE159
C--- SORT THE VECTOR OF KNOT LOCATIONS FROM LEAST TO GREATEST             SPLEE160
C                                                                         SPLEE161
      CALL SORT1 (T,1,K,KX)                                               SPLEE162
C                                                                         SPLEE163
C--- SORT THE VECTOR OF X-VALUES FROM LEAST TO GREATEST AND CARRY         SPLEE164
C--- ALONG THE CORRESPONDING Y-VALUES                                     SPLEE165
C                                                                         SPLEE166
      CALL SORT2 (X,Y,1,N,NX)                                             SPLEE167
C                                                                         SPLEE168
C--- CHECK FOR OBSERVATIONS OUTSIDE KNOT SEQUENCE                         SPLEE169
C                                                                         SPLEE170
      CALL CHECK2 (T,K,KX,X,W,N,NX,NZ,MO)                                 SPLEE171
C                                                                         SPLEE172
C--- COMPUTE NUMBER OF NON-ZERO WEIGHTS                                   SPLEE173
```

```
      NNZ=N-NZ                                                          SPLEE174
C--- DEFINE NEW VECTOR OF KNOTS WITH END POINTS DUPLICATED              SPLEE175
C--- (MD) TIMES                                                         SPLEE176
C                                                                       SPLEE177
      CALL ADKNTS (T,K,KX,MO)                                           SPLEE178
C                                                                       SPLEE179
C--- COMPUTE NUMBER OF B-SPLINES                                        SPLEE180
      NB=K-MO                                                           SPLEE181
C--- COMPUTE NUMBER OF DEGREES OF FREEDOM FOR RESIDUALS                 SPLEE182
      NRSD=NNZ-NB                                                       SPLEE183
      WRITE (6,60) MD,N,NZ,NNZ,K,NB,NY                                  SPLEE184
C--- COMPUTE ESTIMATES OF B-SPLINE COEFFICIENTS                         SPLEE185
C                                                                       SPLEE186
      CALL L2APPR (T,NB,MO,Q,DIAG,BCOEF,JX,K,N,X,Y,W)                   SPLEE187
C                                                                       SPLEE188
C--- COMPUTE UNSCALED VARIANCE-COVARIANCE MATRIX OF                     SPLEE189
C--- B-SPLINE COEFFICIENTS                                              SPLEE190
C                                                                       SPLEE191
      CALL COVAR (KX,NB,JX,MO,Q,XXI)                                    SPLEE192
C                                                                       SPLEE193
C--- COMPUTE (PREDICTED Y-VALUES AND) RESIDUAL STANDARD DEVIATION       SPLEE194
C                                                                       SPLEE195
      CALL RESSD (X,Y,W,N,NX,NKX,NRSD,T,BCOEF,XXI,K,KX,NB,MO,R1,RES,RSD,SPLEE196
     2BIATX,JX,IP)                                                      SPLEE197
C                                                                       SPLEE198
      IF (IP.EQ.0) WRITE (6,50)                                         SPLEE199
      IF (IP.EQ.0) GO TO 120                                            SPLEE200
C--- WRITE B-SPLINE COEFFICIENTS AND THEIR STANDARD DEVIATIONS          SPLEE201
      WRITE (6,20)                                                      SPLEE202
      WRITE (6,30)                                                      SPLEE203
      DO 110 I=1,NB                                                     SPLEE204
      S=RSD*SQRT(XXI(I,I))                                              SPLEE205
      WRITE (6,40) I,BCOEF(I),S                                         SPLEE206
110   CONTINUE                                                          SPLEE207
C--- COMPUTE MULTIPLE CORRELATION COEFFICIENT R-SQUARED (THIS VALUE IS  SPLEE208
C--- NOT PRINTED.  TO PRINT R-SQUARED MAKE A CHANGE IN SUBROUTINE RSQ.) SPLEE209
C                                                                       SPLEE210
120   CALL RSQ (RSD,NRSD,Y,W,N,NX,NNZ)                                  SPLEE211
C                                                                       SPLEE212
C--- CREATE FINE MESH OF EVENLY SPACED X-VALUES BETWEEN END KNOTS       SPLEE213
C--- AND COMPUTE PREDICTED Y-VALUES THERE                               SPLEE214
C                                                                       SPLEE215
      CALL XYFINE (NF,T,BCOEF,K,KX,NB,MO,XF,YF)                         SPLEE216
C                                                                       SPLEE217
C--- COMPUTE STANDARD DEVIATION OF PREDICTED Y-VALUES                   SPLEE218
C                                                                       SPLEE219
      CALL SDYFIN (XF,YFSD,NF,T,K,MO,XXI,KX,RSD,BIATX,JX)               SPLEE220
C                                                                       SPLEE221
C--- COMPUTE CONFIDENCE INTERVALS FOR PREDICTED Y-VALUES USING          SPLEE222
C--- SCHEFFE'S TECHNIQUE (SEE REFERENCE IN SUBROUTINE CIYFIN)           SPLEE223
C                                                                       SPLEE224
      CALL CIYFIN (XF,YF,YFSD,NF,RSD,AL,DL,C,NRSD,NB,YFL,YFU,IP)        SPLEE225
C                                                                       SPLEE226
C--- COMPUTE X CONFIDENCE INTERVALS FOR SPECIFIED Y-VALUES              SPLEE227
C                                                                       SPLEE228
      CALL YTOXCI (XF,YFL,YF,YFU,NF,YY,NY,NYX)                          SPLEE229
C                                                                       SPLEE230
C--- COMPUTE PIECEWISE POLYNOMIAL REPRESENTATION OF SPLINE              SPLEE231
```

```
C                                                                    SPLEE232
      CALL PPREP (T,BCOEF,SCRTCH,DIAG,Q,KX,JX,NB,MO,IP)               SPLEE233
C                                                                    SPLEE234
C--- PLOT KNOT LOCATIONS AND RESIDUALS VS.  INDEPENDENT VARIABLE      SPLEE235
C                                                                    SPLEE236
      CALL PLOTSR (X,N,NX,R1,RES,R2,NKX,T,K,KX)                       SPLEE237
C                                                                    SPLEE238
      WRITE (6,100)                                                   SPLEE239
      RETURN                                                          SPLEE240
      END                                                             SPLEE241
```

CPR*NS(1).XYFINE(1)

```
      SUBROUTINE XYFINE (NF,T,BCOEF,K,KX,NB,MO,XF,YF)                    XYFINE01
C-----------------------------------------------------------------------XYFINE02
C     XYFINE   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING      XYFINE03
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C.  XYFINE04
C              AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION  XYFINE05
C     FOR: CREATING A FINE MESH OF VALUES OVER THE DOMAIN OF X-VALUES    XYFINE06
C          WHERE OBSERVATIONS WERE MADE AND COMPUTING CORRESPONDING      XYFINE07
C          PREDICTED Y-VALUES                                            XYFINE08
C     SUBPROGRAMS CALLED: BVALUE                                         XYFINE09
C     CURRENT VERSION COMPLETED MARCH 19, 1980                           XYFINE10
C-----------------------------------------------------------------------XYFINE11
      DIMENSION T(KX),BCOEF(KX),XF(NF),YF(NF)                            XYFINE12
C---- CREATE FINE MESH OF X VALUES OVER INTERVAL SPANNED BY KNOTS        XYFINE13
10    FORMAT (//5X,13H<<<<< GRID OF,I5,1X,24H EVENLY SPACED X VALUES ,   XYFINE14
     2 13HCREATED >>>>>)                                                 XYFINE15
      C=(T(K)-T(1))/FLOAT(NF-1)                                          XYFINE16
      DO 20 I=1,NF                                                       XYFINE17
      XF(I)=FLOAT(I-1)*C+T(1)                                            XYFINE18
20    CONTINUE                                                           XYFINE19
C---- COMPUTE PREDICTED Y VALUE AT EACH X VALUE                          XYFINE20
      DO 30 I=1,NF                                                       XYFINE21
      XX=XF(I)                                                           XYFINE22
      YF(I)=BVALUE(T,BCOEF,NB,MO,XX,0)                                   XYFINE23
30    CONTINUE                                                           XYFINE24
      WRITE (6,10) NF                                                    XYFINE25
      RETURN                                                             XYFINE26
      END                                                                XYFINE27
```

CPR*NS(1).YTOXCI(8)

```
      SUBROUTINE YTOXCI (XF,YFL,YF,YFU,NF,YY,NY,NYX)                    YTOXC001
C-----------------------------------------------------------------------YTOXC002
C     YTOXCI   WRITTEN BY CHARLES P. REEVE, STATISTICAL ENGINEERING     YTOXC003
C              DIVISION, NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. YTOXC004
C              AS PART OF THE FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION YTOXC005
C     FOR: COMPUTING X CONFIDENCE INTERVALS FOR GIVEN Y-VALUES BY       YTOXC006
C          INVERSE INTERPOLATION ON THE CALIBRATION CURVE AND ITS       YTOXC007
C          UPPER AND LOWER BOUNDS                                       YTOXC008
C     SUBPROGRAMS CALLED: GETX, SORT1                                   YTOXC009
C     CURRENT VERSION COMPLETED SEPTEMBER 3, 1980                       YTOXC010
C-----------------------------------------------------------------------YTOXC011
      DIMENSION XF(NF),YFL(NF),YF(NF),YFU(NF),YY(NYX),IND(6)            YTOXC012
      DATA IND(1),IND(2),IND(3),IND(4),IND(5),IND(6) /1H ,1HS,1HL,1H*,  YTOXC013
     2 1H<,1H>/                                                         YTOXC014
10    FORMAT (/5X,40H<<<<< NO Y-VALUES SPECIFIED FOR INVERSE ,          YTOXC015
     2 19HINTERPOLATION >>>>>)                                          YTOXC016
20    FORMAT (/1X,47H*** LOWER CONFIDENCE CURVE IS NOT MONOTONIC AT ,   YTOXC017
     2 4HYFL(,I4,3H) =,G12.7/5X,21HNO INTERPOLATION DONE)               YTOXC018
30    FORMAT (/1X,45H*** CALIBRATION CURVE IS NOT MONOTONIC AT YF(,I4,  YTOXC019
     2 3H) =,G12.7/5X,21HNO INTERPOLATION DONE)                         YTOXC020
40    FORMAT (/1X,47H*** UPPER CONFIDENCE CURVE IS NOT MONOTONIC AT ,   YTOXC021
     2 4HYFU(,I4,3H) =,G12.7/5X,21HNO INTERPOLATION DONE)               YTOXC022
50    FORMAT (//1X,65(1H-)/1X,29H* COMPUTATION OF CALIBRATION ,         YTOXC023
     2 36HINTERVALS BY INVERSE INTERPOLATION */1X,65(1H-)//24X,         YTOXC024
     3 11HLOWER LIMIT,7X,9HPREDICTED,7X,11HUPPER LIMIT/4X,1HI,6X,4HY(I),YTOXC025
     4 12X,5HFOR X,14X,1HX,14X,5HFOR X/)                                YTOXC026
60    FORMAT (1X,I4,G15.7,3(3X,A1,G13.7))                               YTOXC027
70    FORMAT (//1X,46H*** AT LEAST ONE Y-VALUE IS OUTSIDE THE RANGE ,   YTOXC028
     2 3H***/1X,49H***      OF AT LEAST ONE CALIBRATION CURVE      ***/) YTOXC029
80    FORMAT (/5X,40HS DENOTES THE VALUE OF THE SMALLEST KNOT)          YTOXC030
90    FORMAT (/5X,40HL DENOTES THE VALUE OF THE LARGEST  KNOT)          YTOXC031
100   FORMAT (/5X,41H* DENOTES VALUES OUTSIDE THE RANGE OF THE,         YTOXC032
     2 22H CALIBRATION DATA - NO/7X,29HVALID PREDICTION IS AVAILABLE)   YTOXC033
110   FORMAT (/5X,49H< INDICATES THAT NO VALID LOWER CALIBRATION LIMIT/ YTOXC034
     2 7X,55HGREATER THAN THE MINIMUM POSSIBLE X-VALUE IS AVAILABLE.)   YTOXC035
120   FORMAT (/5X,49H> INDICATES THAT NO VALID UPPER CALIBRATION LIMIT/ YTOXC036
     2 7X,55HSMALLER THAN THE MAXIMUM POSSIBLE X-VALUE IS AVAILABLE.)   YTOXC037
      WRITE (6,50)                                                      YTOXC038
      IF (NY.LT.1) GO TO 230                                            YTOXC039
      M=1                                                               YTOXC040
      IF (YF(1).GT.YF(NF)) M=-1                                         YTOXC041
      NF1=NF-1                                                          YTOXC042
C---  CHECK WHETHER CALIBRATION CURVE AND BOUNDS ARE MONOTONIC          YTOXC043
      DO 150 J=1,NF1                                                    YTOXC044
      D=(YFL(J+1)-YFL(J))*FLOAT(M)                                      YTOXC045
      IF (D.GT.0.0) GO TO 130                                           YTOXC046
      J1=J+1                                                            YTOXC047
      WRITE (6,20) J1,YFL(J1)                                           YTOXC048
      RETURN                                                            YTOXC049
130   D=(YF(J+1)-YF(J))*FLOAT(M)                                        YTOXC050
      IF (D.GT.0.0) GO TO 140                                           YTOXC051
      J1=J+1                                                            YTOXC052
      WRITE (6,30) J1,YF(J1)                                            YTOXC053
      RETURN                                                            YTOXC054
140   D=(YFU(J+1)-YFU(J))*FLOAT(M)                                      YTOXC055
      IF (D.GT.0.0) GO TO 150                                           YTOXC056
      J1=J+1                                                            YTOXC057
```

```
      WRITE (6,40) J1,YFU(J1)                                          YTOXC058
      RETURN                                                           YTOXC059
150   CONTINUE                                                         YTOXC060
C--- ORDER VECTOR OF Y-VALUES FOR WHICH X CONFIDENCE LIMITS ARE TO     YTOXC061
C--- BE COMPUTED                                                       YTOXC062
      CALL SORT1 (YY,1,NY,NYX)                                         YTOXC063
      L1=0                                                             YTOXC064
      L2=0                                                             YTOXC065
      L3=0                                                             YTOXC066
      KS=0                                                             YTOXC067
      KL=0                                                             YTOXC068
C--- IF CURVE IS MONOTONE DECREASING INVERT VECTORS ASSOCIATED         YTOXC069
C--- WITH FINE MESH OF POINTS                                          YTOXC070
      IF (M.EQ.1) GO TO 170                                            YTOXC071
      NHALF=NF/2                                                       YTOXC072
      DO 160 I=1,NHALF                                                 YTOXC073
      J=NF+1-I                                                         YTOXC074
      Q=XF(I)                                                          YTOXC075
      XF(I)=XF(J)                                                      YTOXC076
      XF(J)=Q                                                          YTOXC077
      Q=YFL(I)                                                         YTOXC078
      YFL(I)=YFL(J)                                                    YTOXC079
      YFL(J)=Q                                                         YTOXC080
      Q=YF(I)                                                          YTOXC081
      YF(I)=YF(J)                                                      YTOXC082
      YF(J)=Q                                                          YTOXC083
      Q=YFU(I)                                                         YTOXC084
      YFU(I)=YFU(J)                                                    YTOXC085
      YFU(J)=Q                                                         YTOXC086
160   CONTINUE                                                         YTOXC087
170   DO 220 J=1,NY                                                    YTOXC088
      Y=YY(J)                                                          YTOXC089
C--- GET THREE (3) X-VALUES BY INVERSE INTERPOLATION                   YTOXC090
      CALL GETX (XF,YFL,NF,Y,L1,M,XU,I3,KS,KL)                         YTOXC091
      IF (I3.EQ.1) GO TO 180                                           YTOXC092
      I3=(3-15*M+2*I3+6*M*I3)/2                                        YTOXC093
180   CALL GETX (XF,YF,NF,Y,L2,M,X,I2,KS,KL)                           YTOXC094
      IF (I2.EQ.1) GO TO 190                                           YTOXC095
      I2=4                                                             YTOXC096
      X=0.                                                             YTOXC097
190   CALL GETX (XF,YFU,NF,Y,L3,M,XL,I1,KS,KL)                         YTOXC098
      IF (I1.EQ.1) GO TO 200                                           YTOXC099
      I1=(3+15*M+2*I1-6*M*I1)/2                                        YTOXC100
C--- IF CURVE IS MONOTONE DECREASING REVERSE LIMITS                    YTOXC101
200   IF (M.EQ.1) GO TO 210                                            YTOXC102
      D=XL                                                             YTOXC103
      XL=XU                                                            YTOXC104
      XU=D                                                             YTOXC105
      I=I1                                                             YTOXC106
      I1=I3                                                            YTOXC107
      I3=I                                                             YTOXC108
210   WRITE (6,60) J,Y,IND(I1),XL,IND(I2),X,IND(I3),XU                 YTOXC109
220   CONTINUE                                                         YTOXC110
C--- FLAG Y-VALUES WHICH GIVE INTERPOLATED X-VALUES OUTSIDE THE KNOT   YTOXC111
C--- SPAN                                                              YTOXC112
      IF (KS+KL.EQ.0) RETURN                                           YTOXC113
      WRITE (6,70)                                                     YTOXC114
      WRITE (6,80)                                                     YTOXC115
```

```
      WRITE (6,90)                                                    YTOXC116
      WRITE (6,100)                                                   YTOXC117
      WRITE (6,110)                                                   YTOXC118
      WRITE (6,120)                                                   YTOXC119
      RETURN                                                          YTOXC120
  230 WRITE (6,10)                                                    YTOXC121
      RETURN                                                          YTOXC122
      END                                                             YTOXC123
```

References

[1] Scheffé, H. (1973). A statistical theory of calibration. Annals of Statistics 1, 1-37.

[2] Scheffé, H., J. R. Rosenblatt, and C. H. Spiegelman (1980). Journal of Research of NBS, to appear.

[3] deBoor, Carl (1978). A practical guide to splines. New York: Springer-Verlag.

[4] Smith, Patricia L. (1979). Splines as a useful and convenient statistical tool. The American Statistician 33, 57-62.

[5] Jones, Frank E. (1979). Application of an improved volume calibration system to the calibration of accountability tanks. Nuclear Safeguards Technology 1978, Vol. II, 653-659. International Atomic Energy Agency, Vienna.

[6] Tukey, John W. (1977). Exploratory Data Analysis; Addison-Wesley, London.

[7] Hogben, David, Sally T. Peavy and Ruth N. Varner (1971). OMNITAB II User's Reference Manual. NBS Technical Note 552

[8] Ryder, B. G. (1974). The PFORT Verifier. Software Practice and Experience 4, 359-377.

[9] Spiegelman, C. and K. Eberhardt (1980). Journal of Research of NBS, to appear.

[10] Spiegelman, C. and W. Studden (1980). Journal of Research of NBS, to appear.

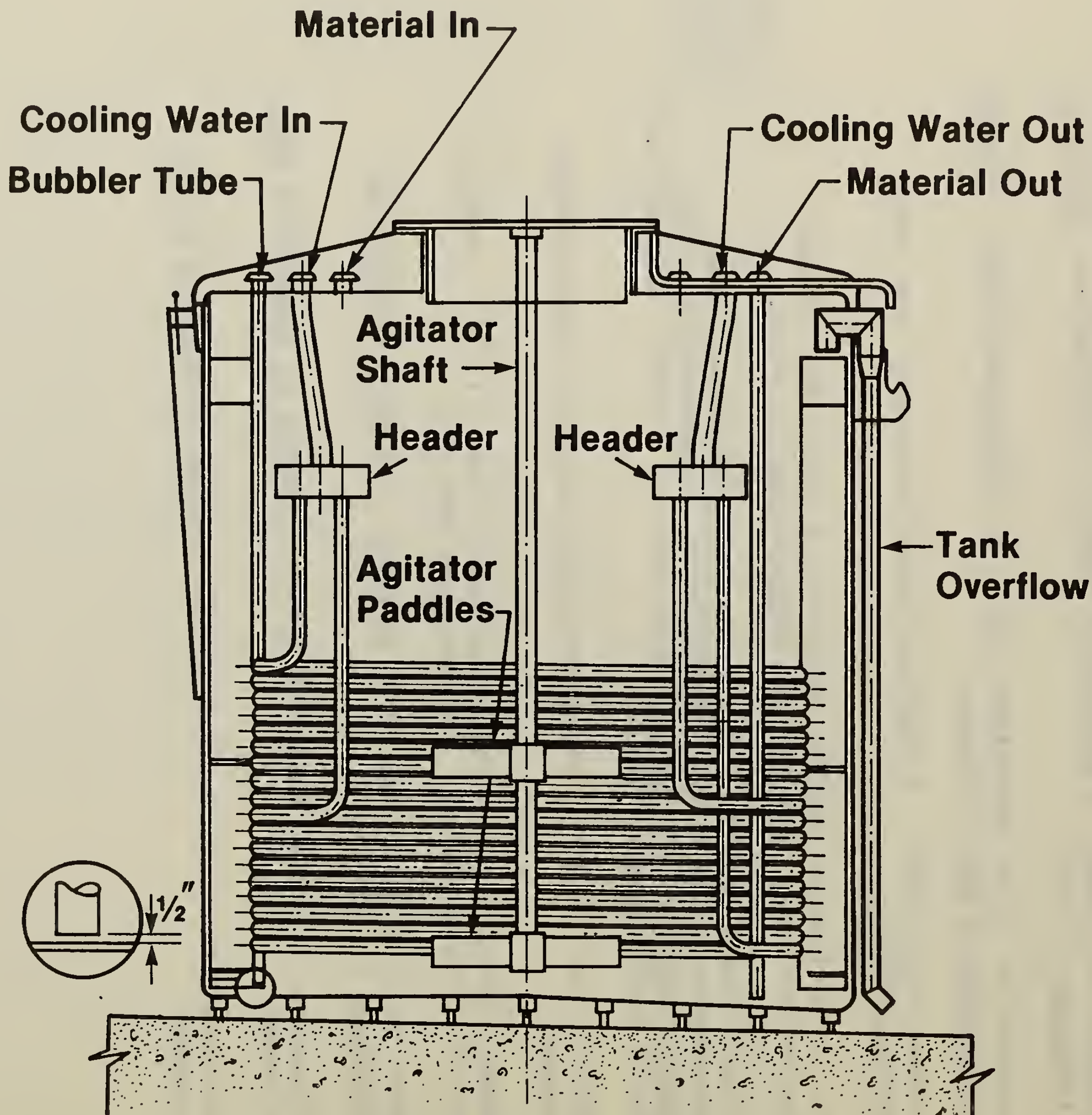

FIG.1. Sketch of an accountability tank.

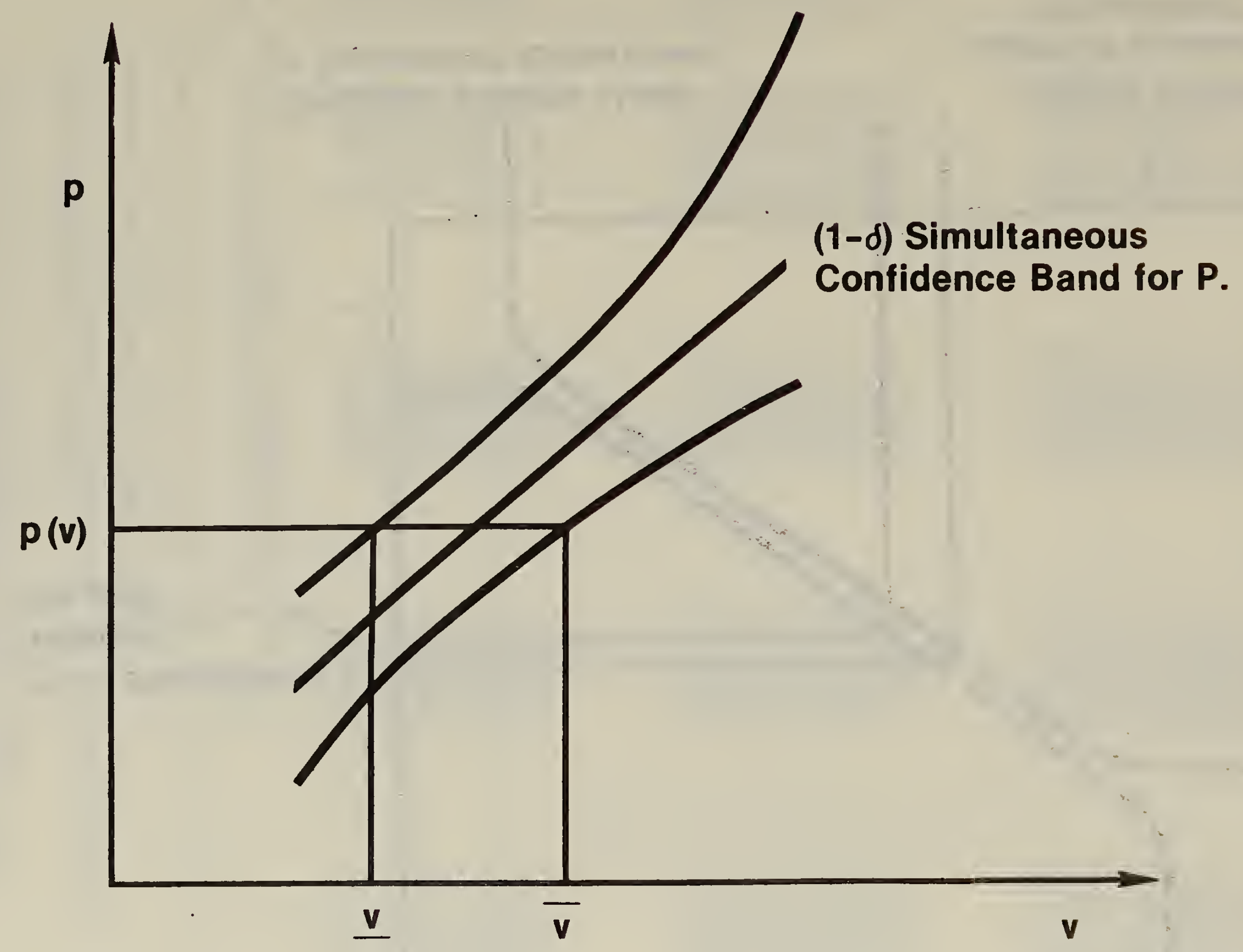

FIG.2. Hypothetical Interval for v obtained from an exact value of p = p(v).

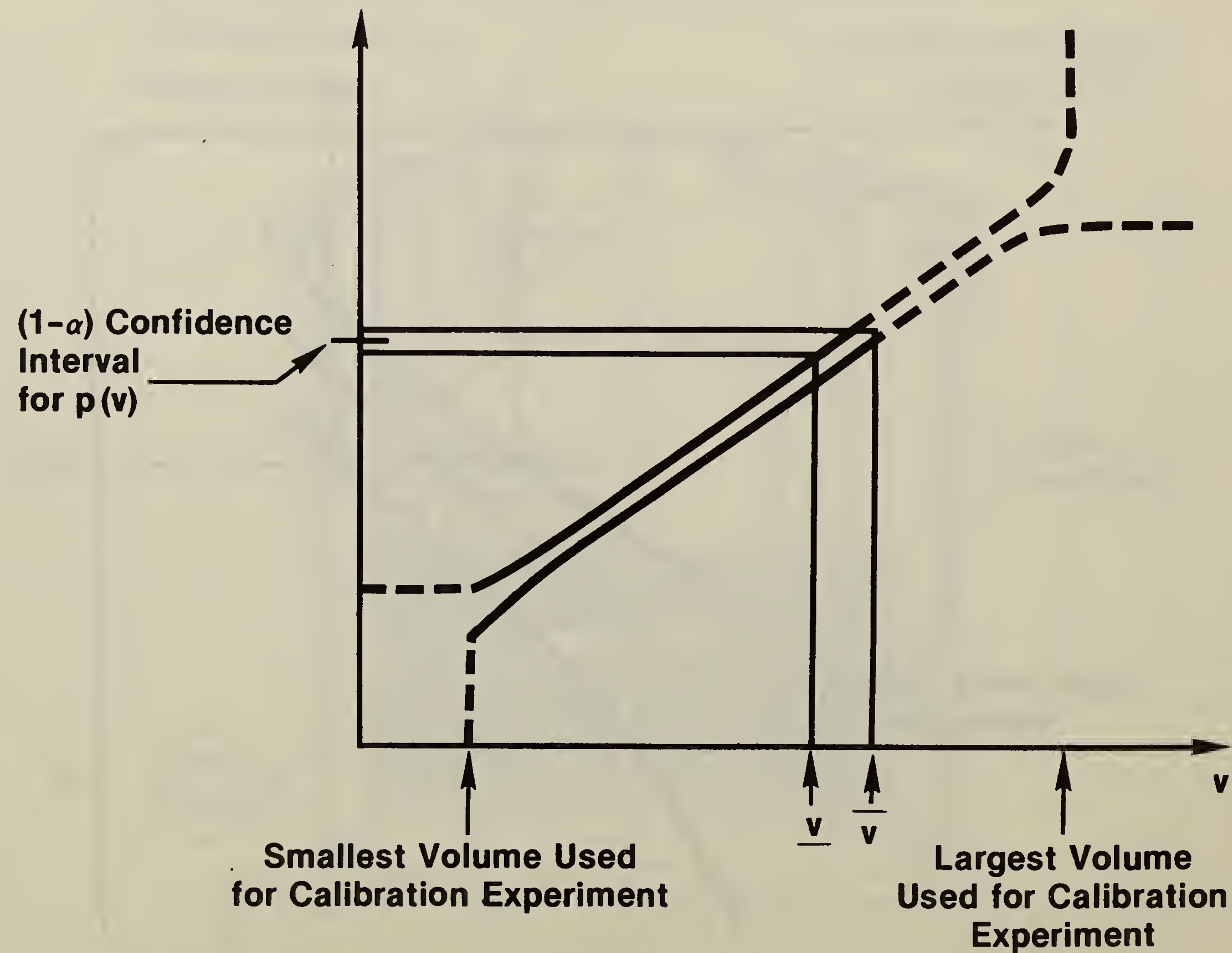

FIG.3. Schematic for approximate construction of the calibration intervals.

* COMPUTATION OF CALIBRATION INTERVALS BY INVERSE INTERPOLATION *

I	Y(I)	LOWER LIMIT FOR X	PREDICTED X	UPPER LIMIT FOR X
1	3713.000	< 1.705250	1.705890	1.709779
2	3823.000	1.749691	1.753002	1.756223
3	3933.000	1.797422	1.800113	1.802745
4	4043.000	1.844952	1.847225	1.849473
5	4153.000	1.892090	1.894297	1.896511
6	4263.000	1.939002	1.941202	1.943401
7	4373.000	1.985893	1.988067	1.990239
8	4483.000	2.032783	2.034931	2.037077
9	4593.000	2.079673	2.081796	2.083916
10	4703.000	2.126563	2.128660	2.130755
.				
.				
.				
.				
164	21643.00	9.704492	9.706814	9.709139
165	21753.00	9.755669	9.758022	9.760378
166	21863.00	9.806846	9.809230	9.811617
167	21973.00	9.858023	9.860439	9.862857
168	22083.00	9.909199	9.911647	9.914098
169	22193.00	9.960375	9.962855	9.965338
170	22303.00	10.01155	10.01406	10.01656
171	22413.00	10.06267	10.06506	10.06742
172	22523.00	10.11361	10.11578	10.11793
173	22633.00	10.16435	10.16651	10.16866
.				
.				
.				
.				
234	29343.00	13.28631	13.28856	13.29082
235	29453.00	13.33740	13.33971	13.34203
236	29563.00	13.38849	13.39087	13.39325
237	29673.00	13.43957	13.44202	13.44447
238	29783.00	13.49066	13.49317	13.49570
239	29893.00	13.54174	13.54433	13.54692
240	30003.00	13.59281	13.59548	13.59815
241	30113.00	L 13.64334	* .0000000	> 13.64334
242	30223.00	L 13.64334	* .0000000	> 13.64334

*** AT LEAST ONE Y-VALUE IS OUTSIDE THE RANGE ***
*** OF AT LEAST ONE CALIBRATION CURVE ***

S DENOTES THE VALUE OF THE SMALLEST KNOT

L DENOTES THE VALUE OF THE LARGEST KNOT

* DENOTES VALUES OUTSIDE THE RANGE OF THE CALIBRATION DATA - NO VALID PREDICTION IS AVAILABLE

< INDICATES THAT NO VALID LOWER CALIBRATION LIMIT GREATER THAN THE MINIMUM POSSIBLE X-VALUE IS AVAILABLE.

> INDICATES THAT NO VALID UPPER CALIBRATION LIMIT SMALLER THAN THE MAXIMUM POSSIBLE X-VALUE IS AVAILABLE.

Figure 4. Calibration chart. Y is pressure in pascals; X is volume in M^3.

```
********************************************
* FIXED-KNOT SPLINE PACKAGE FOR CALIBRATION *
********************************************

REAL DATA FROM A TANK CALIBRATION  (DATA IN FILE CPR*NS.96)

    ----- FULL PRINTOUT -----

    <<<<< EACH END KNOT DUPLICATED  1 TIMES >>>>>

-----------------------------
* SUMMARY OF KNOT LOCATIONS *
-----------------------------

     I       KNOTS(I)

     1       1.70525
     2       1.70325
     3       1.89466
     4       4.54674
     5       5.49482
     6       5.87331
     7       6.44156
     8       7.18990
     9       10.0425
    10       10.2320
    11       12.5044
    12       13.6433
    13       13.6433

-----------------------------------------
* PARAMETERS OF LEAST SQUARES SPLINE FIT *
-----------------------------------------

             DEGREE OF SPLINE =   1

   NUMBER OF      OBSERVATIONS = 172
   NUMBER OF      ZERO WEIGHTS =   0
   NUMBER OF NON-ZERO WEIGHTS = 172
   NUMBER OF             KNOTS =  13
   NUMBER OF         B-SPLINES =  11

          NUMBER OF Y VALUES
      FOR WHICH X CONFIDENCE = 242
    INTERVAL IS TO BE COMPUTED

     <<<<<  11 B-SPLINE COEFFICIENTS COMPUTED >>>>>
```

Figure 5. Preliminaries.

* ANALYSIS OF RESIDUALS *

I	WEIGHT W(I)	X(I)	OBSERVED Y(I)	PREDICTED Y(I)	RESIDUAL(I)	STD DEV OF PREDICTED Y(I)
1	1.0000	1.705250	3711.640	3711.505	.13455	1.052565
2	1.0000	1.705270	3711.420	3711.552	-.13223	1.052453
3	1.0000	1.894010	4152.240	4152.238	.18311-02	.4110981
4	1.0000	1.894140	4152.130	4152.542	-.41168	.4113765
5	1.0000	1.894580	4151.050	4153.569	-2.5190	.4123280
6	1.0000	1.894650	4151.910	4153.732	-1.8225	.4124807
7	1.0000	1.894660	4153.400	4153.756	-.35577	.4125026
8	1.0000	2.084060	4599.320	4598.315	1.0054	.3729185
9	1.0000	2.084110	4600.090	4598.432	1.6581	.3729083
10	1.0000	2.272720	5044.100	5041.136	2.9636	.3359134
11	1.0000	2.273020	5042.730	5041.841	.88947	.3358569
162	1.0000	13.07486	28884.68	28883.46	1.2214	.2921614
163	1.0000	13.26022	29282.92	29282.06	.86230	.3425575
164	1.0000	13.26158	29285.71	29284.98	.72754	.3430482
165	1.0000	13.26232	29284.50	29286.57	-2.0735	.3433159
166	1.0000	13.45198	29695.29	29694.42	.87061	.4235761
167	1.0000	13.45385	29699.14	29698.44	.69897	.4244610
168	1.0000	13.63914	30096.50	30096.89	-.38940	.5180907
169	1.0000	13.64052	30099.62	30099.86	-.23706	.5188241
170	1.0000	13.64133	30098.26	30101.60	-3.3391	.5192548
171	1.0000	13.64143	30102.36	30101.81	.54614	.5193080
172	1.0000	13.64333	30105.91	30105.90	.10254-01	.5203189

RESIDUAL STD DEV	RESIDUAL D.F.
1.48848	161

* ESTIMATION OF B-SPLINE COEFFICIENTS *

I	B-SPLINE COEF	STD DEV
1	3711.5053	1.0525651
2	4153.7559	.41250259
3	10378.705	.41074148
4	12579.214	.57466792
5	13403.155	.74306913
6	14630.321	.62254084
7	16236.443	.41161732
8	22364.084	.44698178
9	22774.945	.43252287
10	27656.846	.39959609
11	30105.922	.52032419

<<<<< GRID OF 300 EVENLY SPACED X VALUES CREATED >>>>>

<<<<< STD. DEV. OF 300 PREDICTED Y VALUES COMPUTED >>>>>

Figure 6. Regression fit.

* PIECEWISE POLYNOMIAL REPRESENTATION OF SPLINES *

	.....INTERVAL.....		COEFFICIENTS OF (X-X(I))**P	
I	X(I)	X(I+1)	P = 0	1
1	1.7052	1.8947	3711.5	2334.9
2	1.8947	4.5467	4153.8	2347.2
3	4.5467	5.4948	10379.	2321.0
4	5.4948	5.8738	12579.	2174.0
5	5.8738	6.4416	13403.	2161.5
6	6.4416	7.1899	14630.	2146.2
7	7.1899	10.043	16236.	2148.1
8	10.043	10.232	22364.	2168.6
9	10.232	12.504	22775.	2148.3
10	12.504	13.643	27657.	2150.4

Figure 7. Ordinary polynomial representation of the fitted spline.

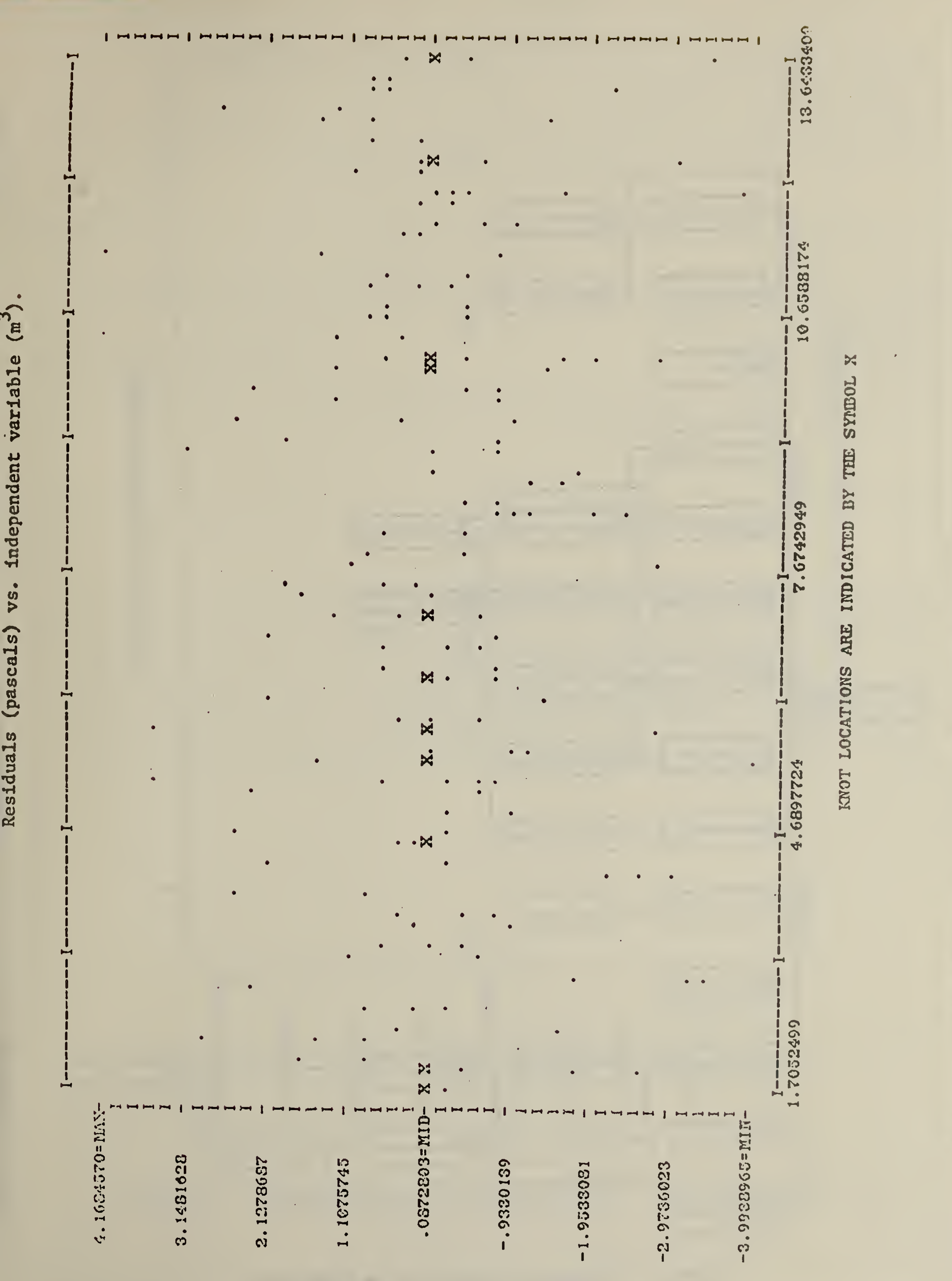

Figure 8. Residual plot.

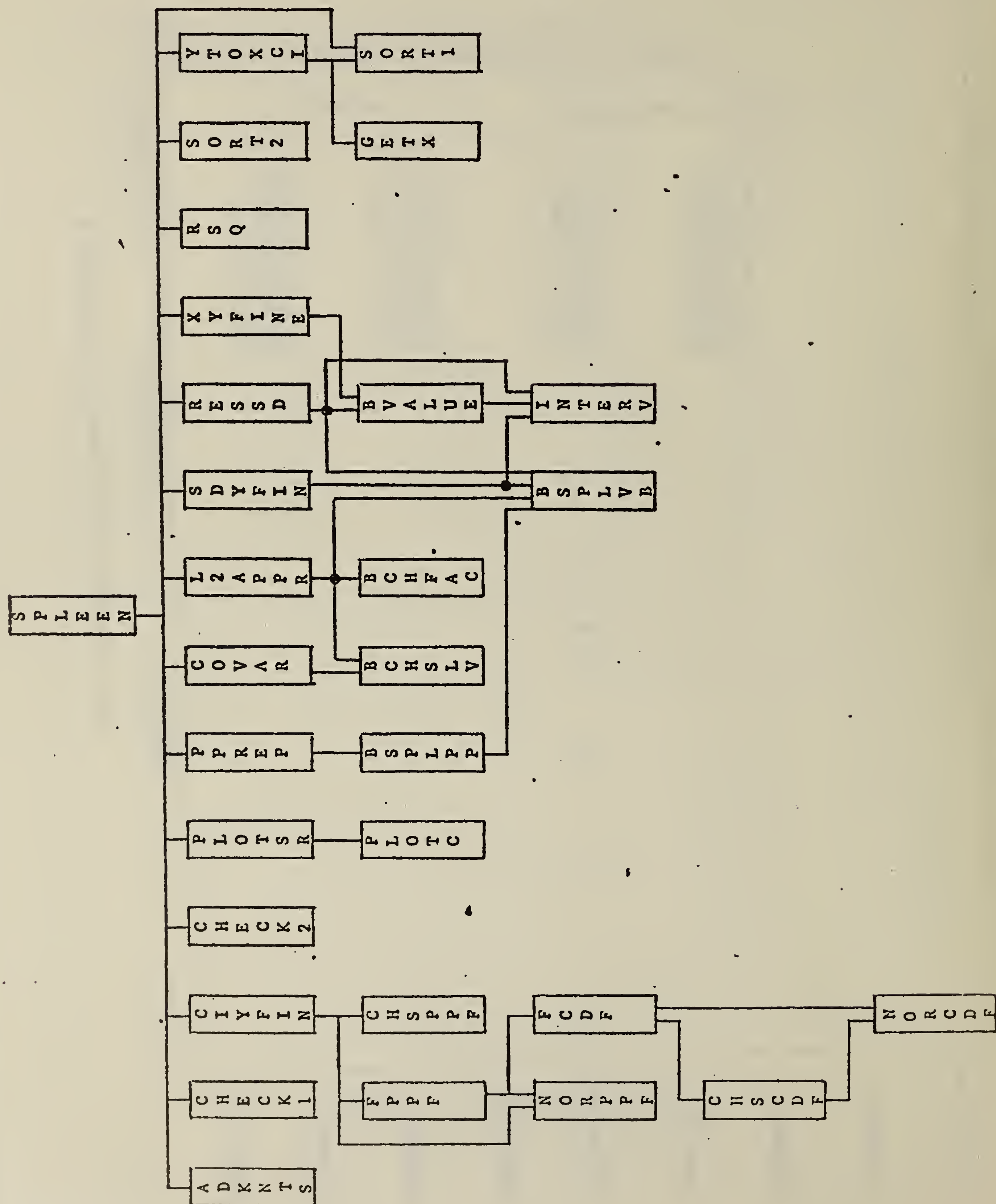

Figure 9. Diagram of subroutine interactions.

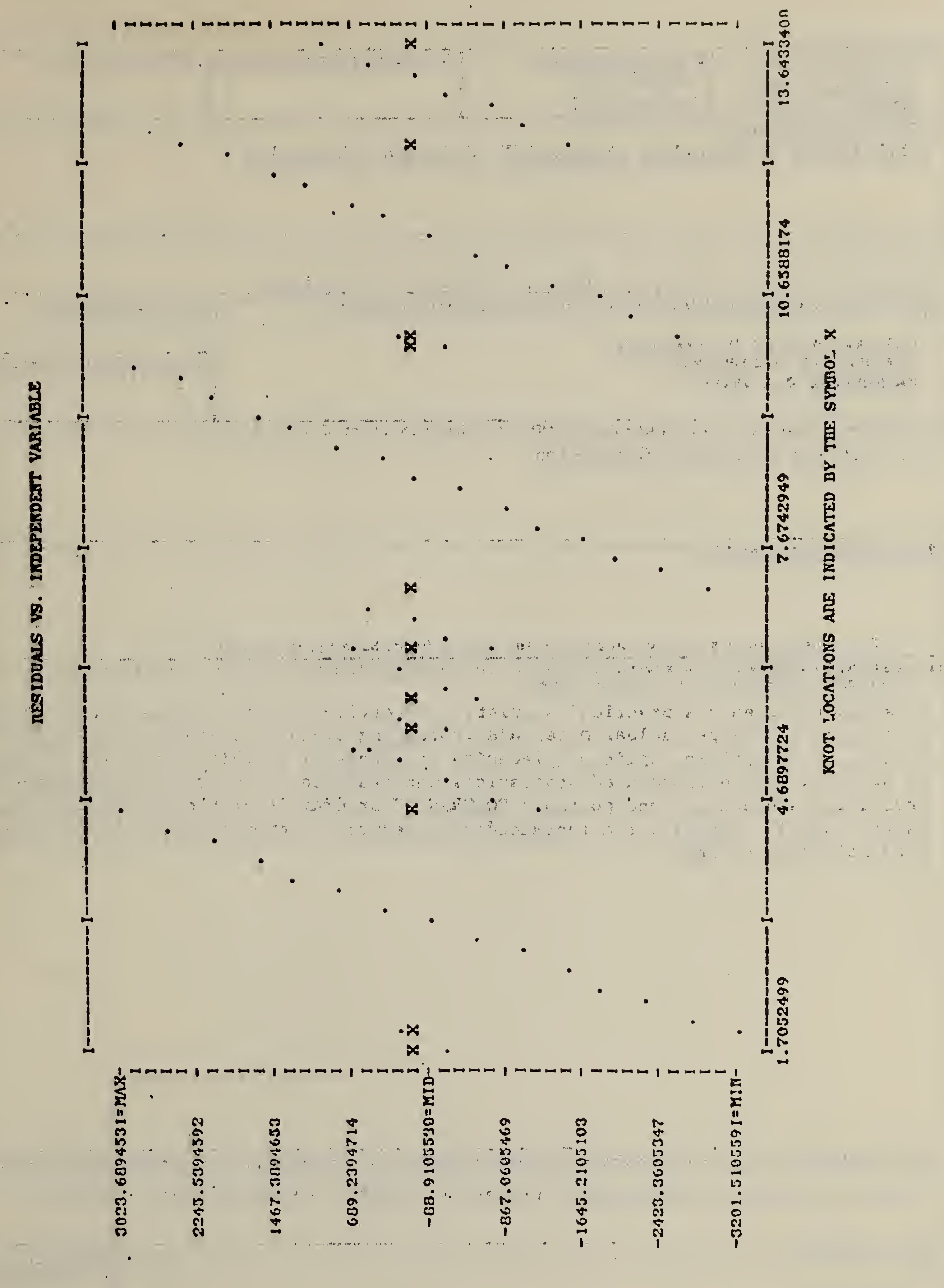

Figure 10. Diagnostic plot of residuals from a zero-degree spline fit (i.e., a step function).

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBS 80-2151	2. Performing Organ. Report No.	3. Publication Date

4. TITLE AND SUBTITLE

A NEW METHOD OF ASSIGNING UNCERTAINTY IN VOLUME CALIBRATION

5. AUTHOR(S)

James A. Lechner, Charles P. Reeve, Clifford H. Spiegelman

6. PERFORMING ORGANIZATION (*If joint or other than NBS, see instructions*)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (*Street, City, State, ZIP*)

U.S. Nuclear Regulatory Commission

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (*A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here*)

This paper presents a practical statistical overview of the pressure volume calibration curve for large nuclear materials processing tanks. It explains the appropriateness of applying splines (piecewise polynomials) to this curve, and it presents an overview of the associated statistical uncertainties. In order to implement these procedures a practical and portable FORTRAN IV program is provided along with its users' manual. Finally, the recommended procedure is demonstrated on actual tank data collected by NBS.

12. KEY WORDS (*Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons*)

Volume calibration; differential pressure; splines; accountability; statistics.

13. AVAILABILITY

[X] Unlimited
[ ] For Official Distribution. Do Not Release to NTIS
[ ] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
[X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

113

15. Price

$9.00