11.4 Least-Squares Cubic Spline Fit

A. Purpose

A cubic spline function with NB − 1 segments is a function consisting of NB − 1 pieces, each of which is a cubic polynomial. At the abscissae, called knots, at which adjacent segments meet, the function has C^2 continuity, *i.e.* continuity in value, first derivative, and second derivative.

Subroutine SC2FIT or DC2FIT will determine the (NB − 1)-segment cubic spline function, with user specified knots, that best fits a set of discrete data in the sense of weighted least-squares, and return the values of the fitted spline curve and its first derivative at the knots. The user can then evaluate the curve, or its first or second derivative, at any argument using the Hermite interpolation subroutine, SHINT or DHINT, of Chapter 12.3.

This software can be used for interpolation by setting the number of knots, NB, to be two less than the number of data points, NXY. Setting NB < NXY − 2 gives least-squares approximation.

For spline fitting with more generality, see Chapter 11.5.

B. Usage

B.1 Program Prototype, Single Precision

INTEGER NXY, NB, LDW, IERR1

REAL X(≥NXY)**, Y**(≥NXY)**, SD**(≥NXY)**, B**(≥NB)**, W**(LDW, 5)**, YKNOT**(≥NB)**, YPKNOT**(≥NB)**, SIGFAC**

Assign values to X(), Y(), SD(), NXY, B(), NB, and LDW.

CALL SC2FIT(X, Y, SD, NXY, B, NB, W, LDW, YKNOT, YPKNOT, SIGFAC, IERR1)

Computed quantities are returned in YKNOT(), YPKNOT(), SIGFAC, and IERR1.

Following the use of SC2FIT the user may use SHINT of Chapter 12.3 to compute values of the fitted curve.

B.2 Argument Definitions

X(), Y() [in] Data pairs (X(I), Y(I), I = 1, ..., NXY). The contents of X() must satisfy X(1) ≤ X(2) ≤ ... ≤ X(NXY).

SD() [in] If SD(1) > 0., each SD(I) must be positive and must be the user's a priori estimate of the standard deviation of the uncertainty (*e.g.*, observational error) in the corresponding data value Y(I). If SD(1) < 0., |SD(1)| will be used as the a priori standard deviation of each data value Y(I). In this case the array SD() may be dimensioned as SD(1).

NXY [in] Number of data points. Require NXY ≥ 4.

B() [in] Set by the user to specify the knot abscissae and endpoints for the spline curve. Must satisfy B(1) < B(2) < ... < B(NB). Also require B(1) ≤ X(1), B(NB) ≥ X(NXY).

NB [in] Number of knots, including endpoints. The number of segments in the spline curve will be NB − 1. The number of degrees of freedom for the fit will be NB + 2. Require 2 ≤ NB ≤ NXY − 2.

W(,) [scratch] Working space, dimensioned W(LDW, 5).

LDW [in] Leading dimension for the work array W(,). LDW must be at least NB + 4, but the execution time is less for larger values of LDW, up to NB+3+k, where k is the largest number of data points lying between any adjacent pair of knots.

YKNOT(), YPKNOT() [out] Arrays, each of length at least NB, in which the subroutine will store a definition of the fitted spline curve as a sequence of values of the curve and its first derivative at the knot abscissae B(i). Letting f denote the fitted curve, the elements of these arrays will be set to

YKNOT(i) = f(B(i)), i = 1, ..., NB

YPKNOT(i) = f'(B(i)), i = 1, ..., NB

SIGFAC [out] Set by the subroutine as a measure of the residual error of the fit. See Section D.

IERR1 [out] Error status indicator. Set on the basis of tests done in SC2FIT as well as error indicators IERR2 set by SBACC and IERR3 set by SBSOL. Zero indicates no errors detected. See Section E for the meaning of nonzero values.

B.3 Modifications for Double Precision

For double precision usage change the REAL statement to DOUBLE PRECISION and change the subroutine name SC2FIT to DC2FIT.

C. Examples and Remarks

Example: Given a set of 12 data pairs (x_i, y_i) compute the uniformly weighted least-squares cubic spline fit to these data using six uniformly spaced breakpoints, including endpoints. After determining the spline function $f(x)$, compute and tabulate the quantities x_i, y_i, $f(x_i)$, and $r_i = y_i - f(x_i)$.

This computation is illustrated by the program DRDC2FIT and the output ODDC2FIT. The fitted curve is determined using DC2FIT, and is evaluated using DHINT of Chapter 12.3.

Interpolation: If all of the data abscissae are distinct, and one wants interpolation, rather than the smoothing effect of least-squares approximation, one can choose interpolation by setting NB = NXY − 2. The NB knot abscissae can be assigned in various ways, but one reasonable way is to set B(1) = X(1), B(i) = X($i+1$), for $i = 2$, ..., NB − 1, and B(NB) = X(NXY).

D. Functional Description

Let knot abscissae $b_1 < b_2 < ... < b_{NB}$ be given. A cubic spline function defined over the interval $[b_1, b_{NB}]$ is a cubic polynomial in each subinterval $[b_i, b_{i+1}]$, $i = 1$, ..., NB − 1, with continuity of the value, first derivative, and second derivative at each internal knot, b_2, ..., b_{NB-1}. The set of all cubic spline functions defined relative to this knot set is a linear space of dimension $d =$ NB + 2. If the knot spacing does not depart severely from uniformity a well conditioned set of basis functions for this space is provided by a particular set of cubic spline functions called B-splines, $p_i(x)$, $i = 1$, ..., d, each of which is nonzero over at most four adjacent subintervals, [1].

The problem data are $\{(x_i, y_i, s_i), i = 1, ..., \text{NXY}\}$, where s_i is the a priori standard deviation of the error in the value y_i. The weighted least-squares curve fitting problem then becomes one of determining coefficients c_j to minimize

$$\rho^2(\mathbf{c}) = \sum_{i=1}^{\text{NXY}} \left[\frac{y_i - \sum_{j=1}^{d} c_j p_j(x_i)}{s_i} \right]^2$$

The matrix formulation of this least-squares problem involves a matrix having a banded form in which at most four elements are nonzero in each row. This least-squares problem is solved using the subroutines of Chapter 4.5.

This problem will have a unique set of solution coefficients, c_j, if NB $\leq$ NXY − 2 and the positioning of the knots is such that there exists an indexing of some set of NB + 2 of the distinct data abscissae, x_i (not necessarily the indexing used in the subprogram) such that $b_{i-3} < x_i < b_{i+1}$ for $i = 1$, ..., NB+2. Here b_{-2}, b_{-1}, and b_0 denote fictitious knots to the left of b_1, and b_{NB+1}, b_{NB+2}, and b_{NB+3} denote fictitious knots to the right of b_{NB}, see [2]. If the solution is not unique, no solution will be given and an error code will be returned as described in Section E.

After determining coefficients, c_j, SC2FIT uses subroutine STRC2C to evaluate the value and first derivative of the fitted curve at the knots. These quantities are returned to the user in the arrays YKNOT() and YPKNOT() as the defining parameters of the fitted curve.

Subroutine SC2BAS is called by both SC2FIT and STRC2C to evaluate B-spline basis functions.

References

1. Carl de Boor, *On calculating with B-splines*, **J. Approx. Theory 6**, 1 (1972) 50–62. Also: C. de Boor, **A Practical Guide to Splines**, Springer-Verlag, 1978, 392 pages.
2. John Rice, *Characterization of Chebyshev approximations by splines*, **SIAM J. on Numer. Anal. 4**, 4 (1967) 557–565.

E. Error Procedures and Restrictions

SC2FIT sets IERR1 and issues error messages based on internal tests as well as propagating error information set in IERR2 by SBACC and in IERR3 by SBSOL. See Chapter 4.5 for the meaning of IERR2 and IERR3. In all cases in which IERR1 is set nonzero, no solution will be computed.

IERR1	Meaning
0	No errors detected.
100	NB < 2 or NXY < NB + 2.
200	B(I) $\geq$ B(I+1) for some I.
300	LDW < NB + 4.
400	X(I −1) > X(I) for some I.
500	B(1) > X(1) or B(NB) < X(NXY).
600	Need larger dimension LDW.
700+IERR2	SBACC set IERR2 $\neq$ 0.
800+IERR2	SBACC set IERR2 $\neq$ 0.
900+IERR2	SBACC set IERR2 $\neq$ 0.
1000+IERR3	SBSOL set IERR3 $\neq$ 0. Indicates singularity.
1100	SD(1) = 0.0.
1200	SD(1) > 0.0, and SD(i) $\leq$ 0.0 for some $i \in$ [2, NXY].

F. Supporting Information

Entry	Required Files
DC2FIT	AMACH, DBACC, DBSOL, DC2BAS, DC2FIT, DERM1, DERV1, DHTCC, DNRM2, DTRC2C, ERFIN, ERMSG, IERM1, IERV1
SC2FIT	AMACH, ERFIN, ERMSG, IERM1, IERV1, SBACC, SBSOL, SC2BAS, SC2FIT, SERM1, SERV1, SHTCC, SNRM2, STRC2C

Original code designed by C. L. Lawson and R. J. Hanson, JPL, 1968. Programmed and modified by Lawson, Hanson, T. Lang, and D. Campbell, JPL, 1968–1974. Adapted to Fortran 77 by Lawson and S. Y. Chiu, July 1987.

DRDC2FIT

```
c       program DRDC2FIT
c>> 1996-07-03 DRDC2FIT Krogh   Changed formats for C conversion.
c>> 1994-10-19 DRDC2FIT Krogh   Changes to use M77CON
c>> 1992-04-27 DRDC2FIT CLL Declared all variables.
c>> 1987-12-09 DRDC2FIT Lawson   Initial Code.
c       Demonstration driver for DC2FIT.
c       C. L. Lawson, JPL, Apr 13 1987, 7/23/87
c       ------------------------------------------------------------------
c--D replaces "?": DR?C2FIT, ?C2FIT, ?HINT
c       ------------------------------------------------------------------
      integer I, NB, NDERIV, NW, NXY
      parameter(NXY = 12, NB = 6, NW = 10)
      external DHINT
      double precision DHINT, X(NXY), Y(NXY), SD(01), B(NB), W(NW,5)
      double precision YKNOT(NB), YPKNOT(NB), R, SIGFAC, YFIT
      integer IERR
      data X / 2.D0, 4.D0, 6.D0, 8.D0,10.D0,12.D0,
     *        14.D0,16.D0,18.D0,20.D0,22.D0,24.D0/
      data Y /2.2D0,4.0D0,5.0D0,4.6D0,2.8D0,2.7D0,
     *        3.8D0,5.1D0,6.1D0,6.3D0,5.0D0,2.0D0/
      data B / 2.0D0, 6.4D0, 10.8D0, 15.2D0, 19.6D0, 24.0D0 /
      data NDERIV / 0 /
      data SD(1) / -1.D0 /
c       ------------------------------------------------------------------
      call DC2FIT(X, Y, SD, NXY, B, NB, W, NW, YKNOT,YPKNOT,SIGFAC,IERR)
      print
     *   '('' DRDC2FIT..  Demo driver for DC2FIT.  Also uses DHINT.'')'
      print '(/5x,''IERR ='',i5,'',   SIGFAC ='',f10.5//''  YKNOT() = '',
     *   6f10.5)', IERR,SIGFAC, (YKNOT(I),I=1,NB)
      print '('' YPKNOT() = '',6f10.5)', (YPKNOT(I),I=1,NB)
      print '(/''   I     X        Y        YFIT   R=Y-YFIT''/)'
      do 10 I=1,NXY
         YFIT= DHINT(X(I), NDERIV, NB, B, YKNOT, YPKNOT)
         R=Y(I)-YFIT
         print '(1x,i2,f6.0,2f9.3,f10.3)',I,X(I),Y(I),YFIT,R
   10 continue
      stop
      end
```

ODDC2FIT

```
DRDC2FIT..  Demo driver for DC2FIT.  Also uses DHINT.

    IERR =      0,  SIGFAC =    0.14664

 YKNOT() =      2.20672    5.13370    2.61122    4.61735    6.30079    1.99475
YPKNOT() =      0.76829   -0.05990   -0.25290    0.71946   -0.10933   -2.07029

  I     X         Y         YFIT   R=Y-YFIT

  1     2.      2.200      2.207     -0.007
  2     4.      4.000      3.958      0.042
  3     6.      5.000      5.111     -0.111
  4     8.      4.600      4.430      0.170
  5    10.      2.800      2.959     -0.159
  6    12.      2.700      2.646      0.054
  7    14.      3.800      3.734      0.066
  8    16.      5.100      5.162     -0.062
  9    18.      6.100      6.132     -0.032
 10    20.      6.300      6.233      0.067
 11    22.      5.000      5.033     -0.033
 12    24.      2.000      1.995      0.005
```