NAG Fortran Library Routine Document

G10ABF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G10ABF fits a cubic smoothing spline for a given smoothing parameter.

2 Specification

```
 SUBROUTINE G10ABF(MODE, WEIGHT, N, X, Y, WT, RHO, YHAT, C, LDC, RSS, DF,
1                  RES, H, WK, IFAIL)
 INTEGER           N, LDC, IFAIL
 real              X(N), Y(N), WT(*), RHO, YHAT(N), C(LDC,3), RSS, DF,
1                  RES(N), H(N), WK(9*N+14)
 CHARACTER*1       MODE, WEIGHT
```

3 Description

G10ABF fits a cubic smoothing spline to a set of n observations (x_i, y_i), for $i = 1, 2, \ldots, n$. The spline provides a flexible smooth function for situations in which a simple polynomial or non-linear regression model is unsuitable.

Cubic smoothing splines arise as the unique real-valued solution function f, with absolutely continuous first derivative and squared-integrable second derivative, which minimises:

$$\sum_{i=1}^{n} w_i(y_i - f(x_i))^2 + \rho \int_{-\infty}^{\infty} (f''(x))^2 \, dx,$$

where w_i is the (optional) weight for the ith observation and ρ is the smoothing parameter. This criterion consists of two parts: the first measures the fit of the curve, and the second the smoothness of the curve. The value of the smoothing parameter ρ weights these two aspects; larger values of ρ give a smoother fitted curve but, in general, a poorer fit. For details of how the cubic spline can be estimated see Hutchinson and de Hoog (1985) and Reinsch (1967).

The fitted values, $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)^T$, and weighted residuals, r_i, can be written as

$$\hat{y} = Hy \quad \text{and} \quad r_i = \sqrt{w_i}(y_i - \hat{y}_i)$$

for a matrix H. The residual degrees of freedom for the spline is trace$(I - H)$ and the diagonal elements of H, h_{ii}, are the leverages.

The parameter ρ can be chosen in a number of ways. The fit can be inspected for a number of different values of ρ. Alternatively the degrees of freedom for the spline, which determines the value of ρ, can be specified, or the (generalised) cross-validation can be minimised to give ρ; see G10ACF for further details.

G10ABF requires the x_i to be strictly increasing. If two or more observations have the same x_i-value then they should be replaced by a single observation with y_i equal to the (weighted) mean of the y-values and weight, w_i, equal to the sum of the weights. This operation can be performed by G10ZAF.

The computation is split into three phases.

(i) Compute matrices needed to fit spline.

(ii) Fit spline for a given value of ρ.

(iii) Compute spline coefficients.

When fitting the spline for several different values of ρ, phase (i) need only be carried out once and then phase (ii) repeated for different values of ρ. If the spline is being fitted as part of an iterative weighted least-squares procedure phases (i) and (ii) have to be repeated for each set of weights. In either case, phase (iii) will often only have to be performed after the final fit has been computed.

The algorithm is based on Hutchinson (1986).

4 References

Hastie T J and Tibshirani R J (1990) *Generalized Additive Models* Chapman and Hall

Hutchinson M F (1986) Algorithm 642: A fast procedure for calculating minimum cross-validation cubic smoothing splines *ACM Trans. Math. Software* **12** 150–153

Hutchinson M F and de Hoog F R (1985) Smoothing noisy data with spline functions *Numer. Math.* **47** 99–106

Reinsch C H (1967) Smoothing by spline functions *Numer. Math.* **10** 177–183

5 Parameters

1: MODE – CHARACTER*1 *Input*

On entry: indicates in which mode the routine is to be used.

If MODE = 'P', initialisation and fitting is performed. This partial fit can be used in an iterative weighted least-squares context where the weights are changing at each call to G10ABF or when the coefficients are not required.

If MODE = 'Q', fitting only is performed. Initialisation must have been performed previously by a call to G10ABF with MODE = 'P'. This quick fit may be called repeatedly with different values of RHO without re-initialisation.

If MODE = 'F', initialisation and full fitting is performed and the function coefficients are calculated.

Constraint: MODE = 'P', 'Q' or 'F'.

2: WEIGHT – CHARACTER*1 *Input*

On entry: indicates whether user-defined weights are to be used.

If WEIGHT = 'W', user-defined weights should be supplied in WT.

If WEIGHT = 'U', the data is treated as unweighted.

Constraint: WEIGHT = 'W' or 'U'.

3: N – INTEGER *Input*

On entry: the number of distinct observations, n.

Constraint: N ≥ 3.

4: X(N) – ***real*** array *Input*

On entry: the distinct and ordered values x_i for $i = 1, 2, \ldots, n$.

Constraint: X$(i) <$ X$(i+1)$, for $i = 1, 2, \ldots, n-1$.

5: Y(N) – ***real*** array *Input*

On entry: the values y_i, for $i = 1, 2, \ldots, n$.

6: WT(*) – ***real*** array *Input*

Note: the dimension of the array WT must be at least 1 if WEIGHT = 'U' and N if WEIGHT = 'W'.

On entry: if WEIGHT = 'W', then WT must contain the n weights. If WEIGHT = 'U', WT is not referenced and unit weights are assumed.

Constraint: if WEIGHT = 'W' then WT$(i) > 0.0$ for $i = 1, 2, \ldots, n$.

7: RHO – ***real*** *Input*

On entry: the smoothing parameter, ρ.

Constraint: RHO ≥ 0.0.

8: YHAT(N) – ***real*** array *Output*

On exit: the fitted values, $\hat{y}_i$, for $i = 1, 2, \ldots, n$.

9: C(LDC,3) – ***real*** array *Input/Output*

On entry: if MODE = 'Q', C must be unaltered from the previous call to G10ABF with MODE = 'P'. Otherwise C need not be set.

On exit: if MODE = 'F', C contains the spline coefficients. More precisely, the value of the spline at t is given by $((\mathrm{C}(i,3) \times d + \mathrm{C}(i,2)) \times d + \mathrm{C}(i,1)) \times d + \hat{y}_i$, where $x_i \leq t < x_{i+1}$ and $d = t - x_i$.

If MODE = 'P' or 'Q', C contains information that will be used in a subsequent call to G10ABF with MODE = 'Q'.

10: LDC – INTEGER *Input*

On entry: the first dimension of the array C as declared in the (sub)program from which G10ABF is called.

Constraint: LDC $\geq$ N $-$ 1.

11: RSS – ***real*** *Output*

On exit: the (weighted) residual sum of squares.

12: DF – ***real*** *Output*

On exit: the residual degrees of freedom.

13: RES(N) – ***real*** array *Output*

On exit: the (weighted) residuals, r_i, for $i = 1, 2, \ldots, n$.

14: H(N) – ***real*** array *Output*

On exit: the leverages, h_{ii}, for $i = 1, 2, \ldots, n$.

15: WK(9*N+14) – ***real*** array *Input/Output*

On entry: if MODE = 'Q', WK must be unaltered from the previous call to G10ABF with MODE = 'P'. Otherwise WK is used as workspace and need not be set.

On exit: if MODE = 'P' or 'Q', WK contains information that will be used in a subsequent call to G10ABF with MODE = 'Q'.

16: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the

value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, N < 3,
or LDC $<$ N $- 1$,
or RHO < 0.0,
or MODE $\neq$ 'Q', 'P' or 'F',
or WEIGHT $\neq$ 'W' or 'U'.

IFAIL $= 2$

On entry, WEIGHT $=$ 'W' and at least one element of WT ≤ 0.0.

IFAIL $= 3$

On entry, X(i) $\geq$ X($i+1$), for some i, $i = 1, 2, \ldots, n-1$.

7 Accuracy

Accuracy depends on the value of ρ and the position of the x values. The values of $x_i - x_{i-1}$ and w_i are scaled and ρ is transformed to avoid underflow and overflow problems.

8 Further Comments

The time taken by the routine is of order n.

Regression splines with a small $(< n)$ number of knots can be fitted by E02BAF and E02BEF.

9 Example

The data, given by Hastie and Tibshirani (1990), is the age, x_i, and C-peptide concentration (pmol/ml), y_i, from a study of the factors affecting insulin-dependent diabetes mellitus in children. The data is input, reduced to a strictly ordered set by G10ZAF and a spline is fitted by G10ABF with $\rho = 10.0$. The fitted values and residuals are printed.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G10ABF Example Program Text
*     Mark 16 Release. NAG Copyright 1992.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, LDC
      PARAMETER        (NMAX=43,LDC=NMAX-1)
*     .. Local Scalars ..
      real             DF, RHO, RSS
      INTEGER          I, IFAIL, N, NORD
      CHARACTER        MODE, WEIGHT
*     .. Local Arrays ..
      real             C(LDC,3), H(NMAX), RES(NMAX), WK(9*NMAX+14),
     +                 WT(NMAX), WWT(NMAX), X(NMAX), XORD(NMAX),
```

```
     +                 Y(NMAX), YHAT(NMAX), YORD(NMAX)
      INTEGER          IWRK(NMAX)
*     .. External Subroutines ..
      EXTERNAL         G10ABF, G10ZAF
*     .. Executable Statements ..
      WRITE (NOUT,*) ' G10ABF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.GT.0 .AND. N.LE.NMAX) THEN
         READ (NIN,*) MODE, WEIGHT
         READ (NIN,*) RHO
         IF (WEIGHT.EQ.'U' .OR. WEIGHT.EQ.'u') THEN
            READ (NIN,*) (X(I),Y(I),I=1,N)
         ELSE
            READ (NIN,*) (X(I),Y(I),WT(I),I=1,N)
         END IF
         IFAIL = 0
*
*        Sort data into increasing X and
*        remove tied observations and weight accordingly
*
         IFAIL = 0
*
         CALL G10ZAF(WEIGHT,N,X,Y,WT,NORD,XORD,YORD,WWT,RSS,IWRK,IFAIL)
*
*        Fit cubic spline
*
         CALL G10ABF(MODE,'W',NORD,XORD,YORD,WWT,RHO,YHAT,C,LDC,RSS,DF,
     +               RES,H,WK,IFAIL)
*
*        Print results
*
         WRITE (NOUT,*)
         WRITE (NOUT,99999) ' RHO  = ', RHO
         WRITE (NOUT,*)
         WRITE (NOUT,99999) ' Residual sum of squares  = ', RSS
         WRITE (NOUT,99999) ' Degrees of freedom       = ', DF
         WRITE (NOUT,*)
         WRITE (NOUT,*) ' Ordered input data     Output results'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '    X       Y           Fitted Values'
         WRITE (NOUT,*)
         DO 20 I = 1, NORD
            WRITE (NOUT,99998) XORD(I), YORD(I), YHAT(I)
   20    CONTINUE
      END IF
      STOP
*
99999 FORMAT (A,F10.3)
99998 FORMAT (1X,2F8.4,8X,F8.4)
      END
```

9.2 Program Data

```
G10ABF Example Program Data
43
'F', 'U'
10.0
 5.2 4.8    8.8 4.1  10.5 5.2  10.6 5.5  10.4 5.0
 1.8 3.4   12.7 3.4  15.6 4.9   5.8 5.6   1.9 3.7
 2.2 3.9    4.8 4.5   7.9 4.8   5.2 4.9   0.9 3.0
11.8 4.6    7.9 4.8  11.5 5.5  10.6 4.5   8.5 5.3
11.1 4.7   12.8 6.6  11.3 5.1   1.0 3.9  14.5 5.7
11.9 5.1    8.1 5.2  13.8 3.7  15.5 4.9   9.8 4.8
11.0 4.4   12.4 5.2  11.1 5.1   5.1 4.6   4.8 3.9
 4.2 5.1    6.9 5.1  13.2 6.0   9.9 4.9  12.5 4.1
13.2 4.6    8.9 4.9  10.8 5.1
```

9.3 Program Results

```
 G10ABF Example Program Results

RHO  =     10.000

Residual sum of squares  =     11.288
Degrees of freedom       =     27.785

 Ordered input data     Output results

    X       Y           Fitted Values

  0.9000  3.0000          3.3674
  1.0000  3.9000          3.4008
  1.8000  3.4000          3.6642
  1.9000  3.7000          3.7016
  2.2000  3.9000          3.8214
  4.2000  5.1000          4.5265
  4.8000  4.2000          4.6471
  5.1000  4.6000          4.7561
  5.2000  4.8500          4.7993
  5.8000  5.6000          5.0458
  6.9000  5.1000          5.1204
  7.9000  4.8000          4.9590
  8.1000  5.2000          4.9262
  8.5000  5.3000          4.8595
  8.8000  4.1000          4.8172
  8.9000  4.9000          4.8095
  9.8000  4.8000          4.8676
  9.9000  4.9000          4.8818
 10.4000  5.0000          4.9445
 10.5000  5.2000          4.9521
 10.6000  5.0000          4.9572
 10.8000  5.1000          4.9613
 11.0000  4.4000          4.9614
 11.1000  4.9000          4.9618
 11.3000  5.1000          4.9623
 11.5000  5.5000          4.9568
 11.8000  4.6000          4.9338
 11.9000  5.1000          4.9251
 12.4000  5.2000          4.8943
 12.5000  4.1000          4.8944
 12.7000  3.4000          4.9051
 12.8000  6.6000          4.9138
 13.2000  5.3000          4.9239
 13.8000  3.7000          4.8930
 14.5000  5.7000          4.9938
 15.5000  4.9000          4.9773
 15.6000  4.9000          4.9682
```