E01BAF – NAG Fortran Library Routine Document

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

E01BAF determines a cubic-spline interpolant to a given set of data.

2 Specification

```
SUBROUTINE E01BAF(M, X, Y, LAMDA, C, LCK, WRK, LWRK, IFAIL)
INTEGER          M, LCK, LWRK, IFAIL
real             X(M), Y(M), LAMDA(LCK), C(LCK), WRK(LWRK)
```

3 Description

This routine determines a cubic spline $s(x)$, defined in the range $x_1 \leq x \leq x_m$, which interpolates (passes exactly through) the set of data points (x_i, y_i), for $i = 1, 2, \ldots, m$, where $m \geq 4$ and $x_1 < x_2 < \ldots < x_m$. Unlike some other spline interpolation algorithms, derivative end conditions are not imposed. The spline interpolant chosen has $m - 4$ interior knots $\lambda_5, \lambda_6, \ldots, \lambda_m$, which are set to the values of $x_3, x_4, \ldots, x_{m-2}$ respectively. This spline is represented in its B-spline form (see Cox [1]):

$$s(x) = \sum_{i=1}^{m} c_i N_i(x),$$

where $N_i(x)$ denotes the normalised B-Spline of degree 3, defined upon the knots $\lambda_i, \lambda_{i+1}, \ldots, \lambda_{i+4}$, and c_i denotes its coefficient, whose value is to be determined by the routine.

The use of B-splines requires eight additional knots λ_1, λ_2, λ_3, λ_4, λ_{m+1}, λ_{m+2}, λ_{m+3} and λ_{m+4} to be specified; the routine sets the first four of these to x_1 and the last four to x_m.

The algorithm for determining the coefficients is as described in [1] except that QR factorization is used instead of LU decomposition. The implementation of the algorithm involves setting up appropriate information for the related routine E02BAF followed by a call of that routine. (For further details of E02BAF, see the routine document.)

Values of the spline interpolant, or of its derivatives or definite integral, can subsequently be computed as detailed in Section 8.

4 References

[1] Cox M G (1975) An algorithm for spline interpolation *J. Inst. Math. Appl.* **15** 95–108

[2] Cox M G (1977) A survey of numerical methods for data and function approximation *The State of the Art in Numerical Analysis* (ed D A H Jacobs) Academic Press 627–668

5 Parameters

1: M — INTEGER *Input*

On entry: m, the number of data points.

Constraint: $M \geq 4$.

2: X(M) — ***real*** array *Input*

On entry: X(i) must be set to x_i, the ith data value of the independent variable x, for $i = 1, 2, \ldots, m$.

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

3: Y(M) — ***real*** array *Input*

On entry: Y(i) must be set to y_i, the ith data value of the dependent variable y, for $i = 1, 2, \ldots, m$.

4: LAMDA(LCK) — ***real*** array *Output*

On exit: the value of λ_i, the ith knot, for $i = 1, 2, \ldots, m + 4$.

5: C(LCK) — ***real*** array *Output*

On exit: the coefficient c_i of the B-spline $N_i(x)$, for $i = 1, 2, \ldots, m$. The remaining elements of the array are not used.

6: LCK — INTEGER *Input*

On entry: the dimension of the arrays LAMDA and C as declared in the (sub)program from which E01BAF is called.

Constraint: $\mathrm{LCK} \geq \mathrm{M} + 4$.

7: WRK(LWRK) — ***real*** array *Workspace*

8: LWRK — INTEGER *Input*

On entry: the dimension of the array WRK as declared in the (sub)program from which E01BAF is called.

Constraint: $\mathrm{LWRK} \geq 6 \times \mathrm{M} + 16$.

9: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, $\mathrm{M} < 4$,
or $\mathrm{LCK} < \mathrm{M} + 4$,
or $\mathrm{LWRK} < 6 \times \mathrm{M} + 16$.

IFAIL = 2

The X-values fail to satisfy the condition

$\mathrm{X}(1) < \mathrm{X}(2) < \mathrm{X}(3) < \ldots < \mathrm{X}(\mathrm{M})$.

7 Accuracy

The rounding errors incurred are such that the computed spline is an exact interpolant for a slightly perturbed set of ordinates $y_i + \delta y_i$. The ratio of the root-mean-square value of the δy_i to that of the y_i is no greater than a small multiple of the relative ***machine precision***.

8 Further Comments

The time taken by the routine is approximately proportional to m.

All the x_i are used as knot positions except x_2 and x_{m-1}. This choice of knots (see Cox [2]) means that $s(x)$ is composed of $m-3$ cubic arcs as follows. If $m = 4$, there is just a single arc space spanning the whole interval x_1 to x_4. If $m \geq 5$, the first and last arcs span the intervals x_1 to x_3 and x_{m-2} to x_m respectively. Additionally if $m \geq 6$, the ith arc, for $i = 2, 3, \ldots, m-4$ spans the interval x_{i+1} to x_{i+2}.

After the call

```
CALL E01BAF (M, X, Y, LAMDA, C, LCK, WRK, LWRK, IFAIL)
```

the following operations may be carried out on the interpolant $s(x)$.

The value of $s(x)$ at $x = \mathrm{XVAL}$ can be provided in the ***real*** variable SVAL by the call

```
CALL E02BBF (M+4, LAMDA, C, XVAL, SVAL, IFAIL)
```

The values of $s(x)$ and its first three derivatives at $x = \mathrm{XVAL}$ can be provided in the ***real*** array SDIF of dimension 4, by the call

```
CALL E02BCF (M+4, LAMDA, C, XVAL, LEFT, SDIF, IFAIL)
```

Here LEFT must specify whether the left- or right-hand value of the third derivative is required (see E02BCF for details).

The value of the integral of $s(x)$ over the range x_1 to x_m can be provided in the ***real*** variable SINT by

```
CALL E02BDF (M+4, LAMDA, C, SINT, IFAIL)
```

9 Example

The example program sets up data from 7 values of the exponential function in the interval 0 to 1. E01BAF is then called to compute a spline interpolant to these data.

The spline is evaluated by E02BBF, at the data points and at points halfway between each adjacent pair of data points, and the spline values and the values of e^x are printed out.

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.

```
*     E01BAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          M, LCK, LWRK
      PARAMETER        (M=7,LCK=M+4,LWRK=6*M+16)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             FIT, XARG
      INTEGER          I, IFAIL, J, R
*     .. Local Arrays ..
      real             C(LCK), LAMDA(LCK), WRK(LWRK), X(M), Y(M)
*     .. External Subroutines ..
      EXTERNAL         E01BAF, E02BBF
```

```
*     .. Intrinsic Functions ..
      INTRINSIC        EXP
*     .. Data statements ..
      DATA             (X(I),I=1,M)/0.0e0, 0.2e0, 0.4e0, 0.6e0, 0.75e0,
     +                 0.9e0, 1.0e0/
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E01BAF Example Program Results'
      DO 20 I = 1, M
         Y(I) = EXP(X(I))
   20 CONTINUE
      IFAIL = 0
*
      CALL E01BAF(M,X,Y,LAMDA,C,LCK,WRK,LWRK,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) '   J    Knot LAMDA(J+2)   B-spline coeff C(J)'
      WRITE (NOUT,*)
      J = 1
      WRITE (NOUT,99998) J, C(1)
      DO 40 J = 2, M - 1
         WRITE (NOUT,99999) J, LAMDA(J+2), C(J)
   40 CONTINUE
      WRITE (NOUT,99998) M, C(M)
      WRITE (NOUT,*)
      WRITE (NOUT,*)
     +  '   R        Abscissa            Ordinate             Spline'
      WRITE (NOUT,*)
      DO 60 R = 1, M
         IFAIL = 0
*
         CALL E02BBF(M+4,LAMDA,C,X(R),FIT,IFAIL)
*
         WRITE (NOUT,99999) R, X(R), Y(R), FIT
         IF (R.LT.M) THEN
            XARG = 0.5e0*(X(R)+X(R+1))
*
            CALL E02BBF(M+4,LAMDA,C,XARG,FIT,IFAIL)
*
            WRITE (NOUT,99997) XARG, FIT
         END IF
   60 CONTINUE
      STOP
*
99999 FORMAT (1X,I4,F15.4,2F20.4)
99998 FORMAT (1X,I4,F35.4)
99997 FORMAT (1X,F19.4,F40.4)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
E01BAF Example Program Results

   J    Knot LAMDA(J+2)   B-spline coeff C(J)

   1                             1.0000
   2         0.0000              1.1336
   3         0.4000              1.3726
   4         0.6000              1.7827
   5         0.7500              2.1744
   6         1.0000              2.4918
   7                             2.7183

   R        Abscissa            Ordinate             Spline

   1         0.0000              1.0000              1.0000
             0.1000                                  1.1052
   2         0.2000              1.2214              1.2214
             0.3000                                  1.3498
   3         0.4000              1.4918              1.4918
             0.5000                                  1.6487
   4         0.6000              1.8221              1.8221
             0.6750                                  1.9640
   5         0.7500              2.1170              2.1170
             0.8250                                  2.2819
   6         0.9000              2.4596              2.4596
             0.9500                                  2.5857
   7         1.0000              2.7183              2.7183
```