E02BBF – 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

E02BBF evaluates a cubic spline from its B-spline representation.

2 Specification

```
SUBROUTINE E02BBF(NCAP7, LAMDA, C, X, S, IFAIL)
INTEGER           NCAP7, IFAIL
real              LAMDA(NCAP7), C(NCAP7), X, S
```

3 Description

This routine evaluates the cubic spline $s(x)$ at a prescribed argument x from its augmented knot set λ_i, for $i = 1, 2, \ldots, n + 7$, (see E02BAF) and from the coefficients c_i, for $i = 1, 2, \ldots, q$ in its B-spline representation

$$s(x) = \sum_{i=1}^{q} c_i N_i(x).$$

Here $q = \bar{n} + 3$, where $\bar{n}$ is the number of intervals of the spline, and $N_i(x)$ denotes the normalised B-spline of degree 3 defined upon the knots $\lambda_i, \lambda_{i+1}, \ldots, \lambda_{i+4}$. The prescribed argument x must satisfy $\lambda_4 \le x \le \lambda_{\bar{n}+4}$.

It is assumed that $\lambda_j \ge \lambda_{j-1}$, for $j = 2, 3, \ldots, \bar{n} + 7$, and $\lambda_{\bar{n}+4} > \lambda_4$.

If x is a point at which 4 knots coincide, $s(x)$ is discontinuous at x; in this case, S contains the value defined as x is approached from the right.

The method employed is that of evaluation by taking convex combinations due to de Boor [4]. For further details of the algorithm and its use see Cox [1] and [3].

It is expected that a common use of E02BBF will be the evaluation of the cubic spline approximations produced by E02BAF. A generalization of E02BBF which also forms the derivative of $s(x)$ is E02BCF. E02BCF takes about 50% longer than E02BBF.

4 References

[1] Cox M G (1972) The numerical evaluation of B-splines *J. Inst. Math. Appl.* **10** 134–149

[2] Cox M G (1978) The numerical evaluation of a spline from its B-spline representation *J. Inst. Math. Appl.* **21** 135–143

[3] Cox M G and Hayes J G (1973) Curve fitting: A guide and suite of algorithms for the non-specialist user *NPL Report NAC 26* National Physical Laboratory

[4] de Boor C (1972) On calculating with B-splines *J. Approx. Theory* **6** 50–62

5 Parameters

1: NCAP7 — INTEGER *Input*

On entry: $\bar{n} + 7$, where $\bar{n}$ is the number of intervals (one greater than the number of interior knots, i.e., the knots strictly within the range λ_4 to $\lambda_{\bar{n}+4}$) over which the spline is defined.

Constraint: NCAP7 ≥ 8.

2: LAMDA(NCAP7) — ***real*** array *Input*

On entry: LAMDA(j) must be set to the value of the jth member of the complete set of knots, λ_j for $j = 1, 2, \ldots, \bar{n} + 7$.

Constraint: the LAMDA(j) must be in non-decreasing order with LAMDA(NCAP7 − 3) > LAMDA(4).

3: C(NCAP7) — ***real*** array *Input*

On entry: the coefficient c_i of the B-spline $N_i(x)$, for $i = 1, 2, \ldots, \bar{n} + 3$. The remaining elements of the array are not used.

4: X — ***real*** *Input*

On entry: the argument x at which the cubic spline is to be evaluated.

Constraint: LAMDA(4) ≤ X ≤ LAMDA(NCAP7 − 3).

5: S — ***real*** *Output*

On exit: the value of the spline, $s(x)$.

6: 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

The argument X does not satisfy LAMDA(4) ≤ X ≤ LAMDA(NCAP7 − 3).

In this case the value of S is set arbitrarily to zero.

IFAIL = 2

NCAP7 < 8, i.e., the number of interior knots is negative.

7 Accuracy

The computed value of $s(x)$ has negligible error in most practical situations. Specifically, this value has an **absolute** error bounded in modulus by $18 \times c_{\max} \times$***machine precision***, where $c_{\max}$ is the largest in modulus of c_j, c_{j+1}, c_{j+2} and c_{j+3}, and j is an integer such that $\lambda_{j+3} \leq x \leq \lambda_{j+4}$. If c_j, c_{j+1}, c_{j+2} and c_{j+3} are all of the same sign, then the computed value of $s(x)$ has a **relative** error not exceeding $20\times$***machine precision*** in modulus. For further details see Cox [2].

8 Further Comments

The time taken by the routine is approximately $\mathrm{C} \times (1 + 0.1 \times \log(\bar{n} + 7))$ seconds, where C is a machine-dependent constant.

Note. The routine does not test all the conditions on the knots given in the description of LAMDA in Section 5, since to do this would result in a computation time approximately linear in $\bar{n} + 7$ instead of $\log(\bar{n} + 7)$. All the conditions are tested in E02BAF, however.

9 Example

Evaluate at 9 equally-spaced points in the interval $1.0 \leq x \leq 9.0$ the cubic spline with (augmented) knots 1.0, 1.0, 1.0, 1.0, 3.0, 6.0, 8.0, 9.0, 9.0, 9.0, 9.0 and normalised cubic B-spline coefficients 1.0, 2.0, 4.0, 7.0, 6.0, 4.0, 3.0.

The example program is written in a general form that will enable a cubic spline with $\bar{n}$ intervals, in its normalised cubic B-spline form, to be evaluated at m equally-spaced points in the interval $\mathrm{LAMDA}(4) \leq x \leq \mathrm{LAMDA}(\bar{n}+4)$. The program is self-starting in that any number of data sets may be supplied.

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.

```
*     E02BBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NC7MAX
      PARAMETER        (NC7MAX=200)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      real             A, B, S, X
      INTEGER          IFAIL, J, M, NCAP, R
*     .. Local Arrays ..
      real             C(NC7MAX), LAMDA(NC7MAX)
*     .. External Subroutines ..
      EXTERNAL         E02BBF
*     .. Intrinsic Functions ..
      INTRINSIC        real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02BBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=80) M
      IF (M.GT.0) THEN
         READ (NIN,*) NCAP
         IF (NCAP+7.LE.NC7MAX) THEN
            READ (NIN,*) (LAMDA(J),J=1,NCAP+7)
            READ (NIN,*) (C(J),J=1,NCAP+3)
            A = LAMDA(4)
            B = LAMDA(NCAP+4)
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        '  J       LAMDA(J)    B-spline coefficient (J-2)'
            WRITE (NOUT,*)
            DO 40 J = 1, NCAP + 7
               IF (J.LT.3 .OR. J.GT.NCAP+5) THEN
                  WRITE (NOUT,99999) J, LAMDA(J)
               ELSE
                  WRITE (NOUT,99999) J, LAMDA(J), C(J-2)
               END IF
   40       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        '  R       Argument      Value of cubic spline'
            WRITE (NOUT,*)
```

```
            DO 60 R = 1, M
               X = (real(M-R)*A+real(R-1)*B)/real(M-1)
               IFAIL = 0
*
               CALL E02BBF(NCAP+7,LAMDA,C,X,S,IFAIL)
*
               WRITE (NOUT,99999) R, X, S
   60       CONTINUE
            GO TO 20
         END IF
      END IF
   80 STOP
*
99999 FORMAT (1X,I3,F14.4,F21.4)
      END
```

9.2 Program Data

```
E02BBF Example Program Data
  9
   4
     1.00
     1.00
     1.00
     1.00
     3.00
     6.00
     8.00
     9.00
     9.00
     9.00
     9.00
     1.00
     2.00
     4.00
     7.00
     6.00
     4.00
     3.00
```

9.3 Program Results

```
 E02BBF Example Program Results

   J       LAMDA(J)    B-spline coefficient (J-2)

   1        1.0000
   2        1.0000
   3        1.0000              1.0000
   4        1.0000              2.0000
   5        3.0000              4.0000
   6        6.0000              7.0000
   7        8.0000              6.0000
   8        9.0000              4.0000
   9        9.0000              3.0000
  10        9.0000
  11        9.0000
```

```
R       Argument      Value of cubic spline

1        1.0000              1.0000
2        2.0000              2.3779
3        3.0000              3.6229
4        4.0000              4.8327
5        5.0000              5.8273
6        6.0000              6.3571
7        7.0000              6.1905
8        8.0000              5.1667
9        9.0000              3.0000
```