NAG Library Routine Document

E02BBF

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 (KIND=nag_wp) LAMDA(NCAP7), C(NCAP7), X, S
```

3 Description

E02BBF 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 normalized B-spline of degree 3 defined upon the knots $\lambda_i, \lambda_{i+1}, \ldots, \lambda_{i+4}$. The prescribed argument x must satisfy $\lambda_4 \leq x \leq \lambda_{\bar{n}+4}$.

It is assumed that $\lambda_j \geq \lambda_{j-1}$, 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 (1972). For further details of the algorithm and its use see Cox (1972) and Cox and Hayes (1973).

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

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

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

Cox M G and Hayes J G (1973) Curve fitting: a guide and suite of algorithms for the non-specialist user *NPL Report NAC26* National Physical Laboratory

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

5 Arguments

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 (KIND=nag_wp) 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 nondecreasing order with LAMDA(NCAP7 − 3) > LAMDA(4).

3: C(NCAP7) – REAL (KIND=nag_wp) 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 referenced.

4: X – REAL (KIND=nag_wp) *Input*

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

Constraint: LAMDA(4) $\leq$ X $\leq$ LAMDA(NCAP7 − 3).

5: S – REAL (KIND=nag_wp) *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. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

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, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $=0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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 $=1$

The argument X does not satisfy LAMDA(4) $\leq$ X $\leq$ 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.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

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 $18 \times c_{\text{max}} \times$ ***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 (1978).

8 Parallelism and Performance

E02BBF is not threaded in any implementation.

9 Further Comments

The time taken is approximately $\text{C} \times (1 + 0.1 \times \log(\bar{n} + 7))$ 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.

10 Example

Evaluate at nine 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 normalized 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 normalized cubic B-spline form, to be evaluated at m equally-spaced points in the interval LAMDA$(4) \leq x \leq$ LAMDA$(\bar{n} + 4)$. The program is self-starting in that any number of datasets may be supplied.

10.1 Program Text

```
    Program e02bbfe

!     E02BBF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: e02bbf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
```

```
!     .. Local Scalars ..
      Real (Kind=nag_wp)                 :: a, b, s, x
      Integer                            :: ifail, j, m, ncap, ncap7, r
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable    :: c(:), lamda(:)
!     .. Intrinsic Procedures ..
      Intrinsic                          :: real
!     .. Executable Statements ..
      Write (nout,*) 'E02BBF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Read (nin,*) m
      Read (nin,*) ncap
      ncap7 = ncap + 7
      Allocate (lamda(ncap7),c(ncap7))

      Read (nin,*) lamda(1:ncap7)
      Read (nin,*) c(1:ncap+3)

      a = lamda(4)
      b = lamda(ncap+4)

      Do r = 1, m
        x = (real(m-r,kind=nag_wp)*a+real(r-1,kind=nag_wp)*b)/                 &
          real(m-1,kind=nag_wp)

        ifail = 0
        Call e02bbf(ncap7,lamda,c,x,s,ifail)

        If (r==1) Then
          Write (nout,*)
          Write (nout,*) '  J       LAMDA(J)    B-spline coefficient (J-2)'
          Write (nout,*)

          Do j = 1, ncap7

            If (j<3 .Or. j>ncap+5) Then
              Write (nout,99999) j, lamda(j)
            Else
              Write (nout,99999) j, lamda(j), c(j-2)
            End If

          End Do

          Write (nout,*)
          Write (nout,*) '  R       Argument      Value of cubic spline'
          Write (nout,*)
        End If

        Write (nout,99999) r, x, s
      End Do

99999 Format (1X,I3,F14.4,F21.4)
    End Program e02bbfe
```

10.2 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
```

10.3 Program Results

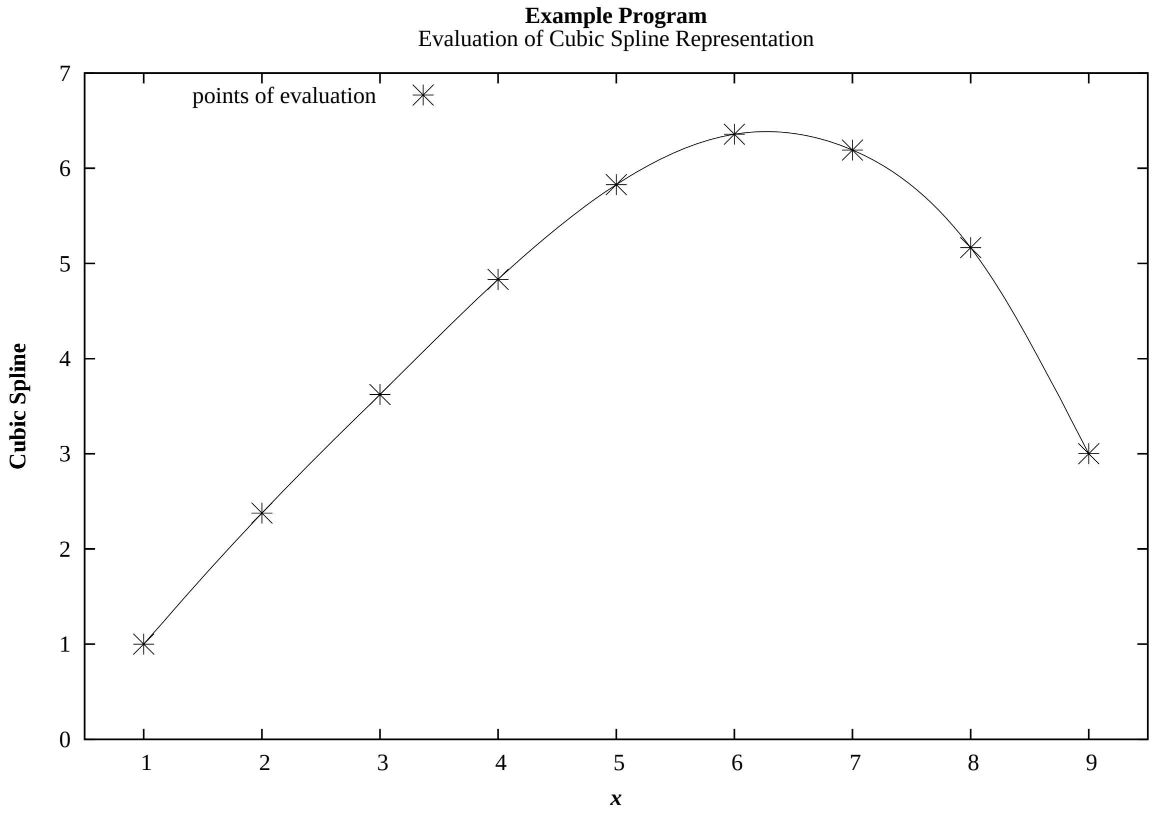
```
 E02BBF Example Program Results

   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
```


Example Program
Evaluation of Cubic Spline Representation
points of evaluation
Cubic Spline
x
0
1
2
3
4
5
6
7
8
9