NAG Library Routine Document

E02BCF

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

E02BCF evaluates a cubic spline and its first three derivatives from its B-spline representation.

2 Specification

```
SUBROUTINE E02BCF (NCAP7, LAMDA, C, X, LEFT, S, IFAIL)
INTEGER            NCAP7, LEFT, IFAIL
REAL (KIND=nag_wp) LAMDA(NCAP7), C(NCAP7), X, S(4)
```

3 Description

E02BCF evaluates the cubic spline $s(x)$ and its first three derivatives at a prescribed argument x. It is assumed that $s(x)$ is represented in terms of its B-spline coefficients c_i, for $i = 1, 2, \ldots, \bar{n} + 3$ and (augmented) ordered knot set λ_i, for $i = 1, 2, \ldots, \bar{n} + 7$, (see E02BAF), i.e.,

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

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

$$\lambda_4 \leq x \leq \lambda_{\bar{n}+4}.$$

At a simple knot λ_i (i.e., one satisfying $\lambda_{i-1} < \lambda_i < \lambda_{i+1}$), the third derivative of the spline is in general discontinuous. At a multiple knot (i.e., two or more knots with the same value), lower derivatives, and even the spline itself, may be discontinuous. Specifically, at a point $x = u$ where (exactly) r knots coincide (such a point is termed a knot of multiplicity r), the values of the derivatives of order $4 - j$, for $j = 1, 2, \ldots, r$, are in general discontinuous. (Here $1 \leq r \leq 4$; $r > 4$ is not meaningful.) You must specify whether the value at such a point is required to be the left- or right-hand derivative.

The method employed is based upon:

(i) carrying out a binary search for the knot interval containing the argument x (see Cox (1978)),

(ii) evaluating the nonzero B-splines of orders 1, 2, 3 and 4 by recurrence (see Cox (1972) and Cox (1978)),

(iii) computing all derivatives of the B-splines of order 4 by applying a second recurrence to these computed B-spline values (see de Boor (1972)),

(iv) multiplying the fourth-order B-spline values and their derivative by the appropriate B-spline coefficients, and summing, to yield the values of $s(x)$ and its derivatives.

E02BCF can be used to compute the values and derivatives of cubic spline fits and interpolants produced by E02BAF.

If only values and not derivatives are required, E02BBF may be used instead of E02BCF, which 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

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 of the spline (which is 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$(\text{NCAP7} - 3) >$ 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 and its derivatives are to be evaluated.

Constraint: LAMDA$(4) \leq$ X $\leq$ LAMDA$(\text{NCAP7} - 3)$.

5: LEFT – INTEGER *Input*

On entry: specifies whether left- or right-hand values of the spline and its derivatives are to be computed (see Section 3). Left- or right-hand values are formed according to whether LEFT is equal or not equal to 1.

If x does not coincide with a knot, the value of LEFT is immaterial.

If $x =$ LAMDA(4), right-hand values are computed.

If $x =$ LAMDA$(\text{NCAP7} - 3)$, left-hand values are formed, regardless of the value of LEFT.

6: S(4) – REAL (KIND=nag_wp) array *Output*

On exit: S(j) contains the value of the $(j-1)$th derivative of the spline at the argument x, for $j = 1, 2, 3, 4$. Note that S(1) contains the value of the spline.

7: 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 $= 1$

NCAP7 < 8, i.e., the number of intervals is not positive.

IFAIL $= 2$

Either LAMDA(4) $\geq$ LAMDA(NCAP7 $- 3$), i.e., the range over which $s(x)$ is defined is null or negative in length, or X is an invalid argument, i.e., X $<$ LAMDA(4) or X $>$ LAMDA(NCAP7 $- 3$).

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 $18 \times c_{\max} \times$ ***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 **relative** error bounded by $20 \times$ ***machine precision***. For full details see Cox (1978).

No complete error analysis is available for the computation of the derivatives of $s(x)$. However, for most practical purposes the absolute errors in the computed derivatives should be small.

8 Parallelism and Performance

E02BCF is not threaded in any implementation.

9 Further Comments

The time taken is approximately linear in $\log(\bar{n} + 7)$.

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

Compute, at the 7 arguments $x = 0$, 1, 2, 3, 4, 5, 6, the left- and right-hand values and first 3 derivatives of the cubic spline defined over the interval $0 \leq x \leq 6$ having the 6 interior knots $x = 1$, 3, 3, 3, 4, 4, the 8 additional knots 0, 0, 0, 0, 6, 6, 6, 6, and the 10 B-spline coefficients 10, 12, 13, 15, 22, 26, 24, 18, 14, 12.

The input data items (using the notation of Section 5) comprise the following values in the order indicated:

$\bar{n}$	m
LAMDA(j),	for $j = 1, 2, \ldots,$ NCAP7
C(j),	for $j = 1, 2, \ldots,$ NCAP7 $- 4$
X(i),	for $i = 1, 2, \ldots, m$

This example program is written in a general form that will enable the values and derivatives of a cubic spline having an arbitrary number of knots to be evaluated at a set of arbitrary points. Any number of datasets may be supplied. The only changes required to the program relate to the dimensions of the arrays LAMDA and C.

10.1 Program Text

```
    Program e02bcfe

!     E02BCF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: e02bcf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: x
      Integer                          :: i, ifail, l, left, m, ncap, ncap7
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: c(:), lamda(:)
      Real (Kind=nag_wp)               :: s(4)
!     .. Executable Statements ..
      Write (nout,*) 'E02BCF Example Program Results'

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

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

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

      Do i = 1, m
        Read (nin,*) x

        Do left = 1, 2

          ifail = 0
          Call e02bcf(ncap7,lamda,c,x,left,s,ifail)

          If (left==1) Then

            If (i==1) Then
              Write (nout,*)
              Write (nout,*)                                                  &
                '      X             Spline    1st deriv   2nd deriv   ',      &
```

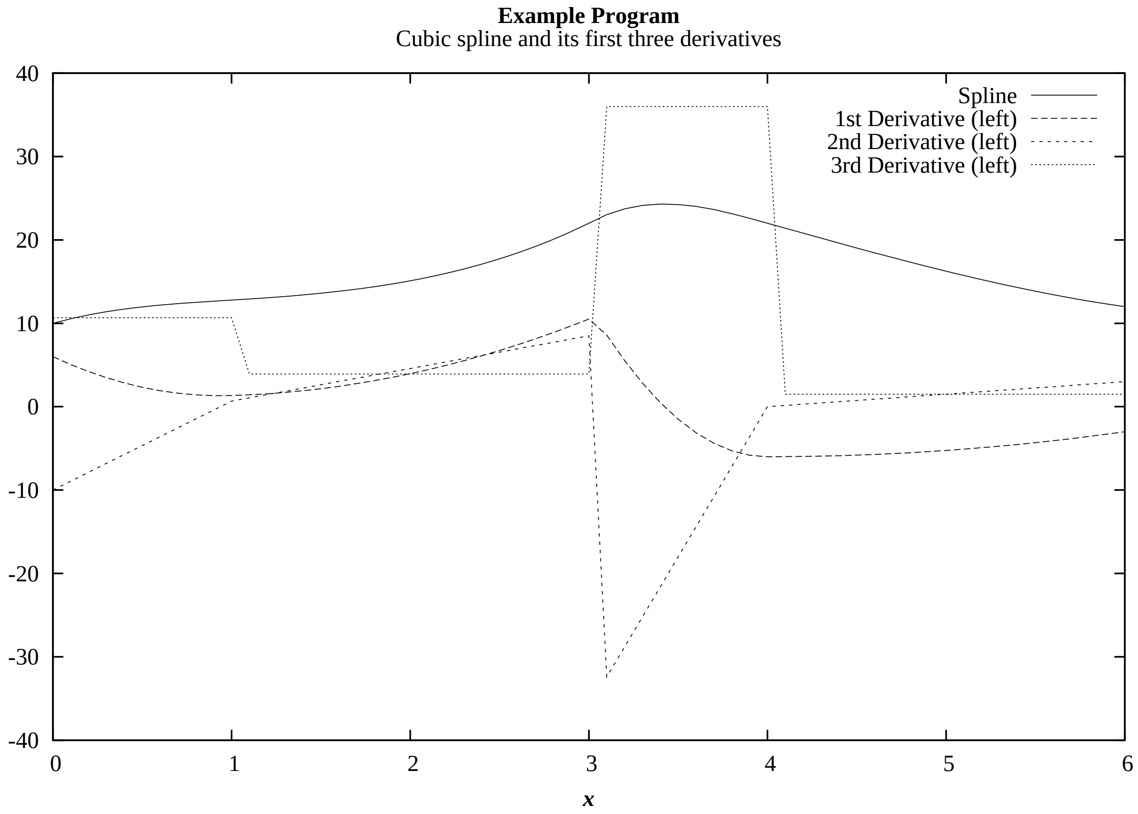
```
                '3rd deriv'
            End If

            Write (nout,*)
            Write (nout,99999) x, '  LEFT', (s(l),l=1,4)
          Else
            Write (nout,99999) x, ' RIGHT', (s(l),l=1,4)
          End If

        End Do

      End Do

99999 Format (1X,E10.2,A,4E12.4)
    End Program e02bcfe
```

10.2 Program Data

```
E02BCF Example Program Data
   7   7
  0.0      0.0      0.0      0.0      1.0      3.0      3.0      3.0
  4.0      4.0      6.0      6.0      6.0      6.0
 10.0     12.0     13.0     15.0     22.0     26.0     24.0     18.0
 14.0     12.0
  0.0
  1.0
  2.0
  3.0
  4.0
  5.0
  6.0
```

10.3 Program Results

```
 E02BCF Example Program Results

       X             Spline    1st deriv   2nd deriv   3rd deriv

   0.00E+00  LEFT  0.1000E+02  0.6000E+01 -0.1000E+02  0.1067E+02
   0.00E+00 RIGHT  0.1000E+02  0.6000E+01 -0.1000E+02  0.1067E+02

   0.10E+01  LEFT  0.1278E+02  0.1333E+01  0.6667E+00  0.1067E+02
   0.10E+01 RIGHT  0.1278E+02  0.1333E+01  0.6667E+00  0.3917E+01

   0.20E+01  LEFT  0.1510E+02  0.3958E+01  0.4583E+01  0.3917E+01
   0.20E+01 RIGHT  0.1510E+02  0.3958E+01  0.4583E+01  0.3917E+01

   0.30E+01  LEFT  0.2200E+02  0.1050E+02  0.8500E+01  0.3917E+01
   0.30E+01 RIGHT  0.2200E+02  0.1200E+02 -0.3600E+02  0.3600E+02

   0.40E+01  LEFT  0.2200E+02 -0.6000E+01  0.0000E+00  0.3600E+02
   0.40E+01 RIGHT  0.2200E+02 -0.6000E+01  0.0000E+00  0.1500E+01

   0.50E+01  LEFT  0.1625E+02 -0.5250E+01  0.1500E+01  0.1500E+01
   0.50E+01 RIGHT  0.1625E+02 -0.5250E+01  0.1500E+01  0.1500E+01

   0.60E+01  LEFT  0.1200E+02 -0.3000E+01  0.3000E+01  0.1500E+01
   0.60E+01 RIGHT  0.1200E+02 -0.3000E+01  0.3000E+01  0.1500E+01
```



Example Program
Cubic spline and its first three derivatives
Spline
1st Derivative (left)
2nd Derivative (left)
3rd Derivative (left)
40
30
20
10
0
-10
-20
-30
-40
0
1
2
3
4
5
6
x