NAG Library Routine Document

E02BDF

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

E02BDF computes the definite integral of a cubic spline from its B-spline representation.

2 Specification

```
SUBROUTINE E02BDF (NCAP7, LAMDA, C, DINT, IFAIL)

INTEGER            NCAP7, IFAIL
REAL (KIND=nag_wp) LAMDA(NCAP7), C(NCAP7), DINT
```

3 Description

E02BDF computes the definite integral of the cubic spline $s(x)$ between the limits $x = a$ and $x = b$, where a and b are respectively the lower and upper limits of the range over which $s(x)$ is defined. 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 $i = 1, 2, \ldots, \bar{n} + 7$, with $\lambda_i = a$, for $i = 1, 2, 3, 4$ and $\lambda_i = b$, for $i = \bar{n} + 4, \ldots, \bar{n} + 7$, (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 method employed uses the formula given in Section 3 of Cox (1975).

E02BDF can be used to determine the definite integrals of cubic spline fits and interpolants produced by E02BAF.

4 References

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

5 Parameters

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 a to b) 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) and satisfy LAMDA(1) $=$ LAMDA(2) $=$ LAMDA(3) $=$ LAMDA(4) and LAMDA(NCAP7 $-$ 3) $=$ LAMDA(NCAP7 $-$ 2) $=$ LAMDA(NCAP7 $-$ 1) $=$ LAMDA(NCAP7).

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: DINT – REAL (KIND=nag_wp) *Output*

On exit: the value of the definite integral of $s(x)$ between the limits $x = a$ and $x = b$, where $a = \lambda_4$ and $b = \lambda_{\bar{n}+4}$.

5: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 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.**

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 $= 2$

At least one of the following restrictions on the knots is violated:

LAMDA(NCAP7 $- 3$) $>$ LAMDA(4),

LAMDA(j) $\geq$ LAMDA($j - 1$),

for $j = 2, 3, \ldots,$ NCAP7, with equality in the cases $j = 2, 3, 4,$ NCAP7 $- 2$, NCAP7 $- 1$, and NCAP7.

7 Accuracy

The rounding errors are such that the computed value of the integral is exact for a slightly perturbed set of B-spline coefficients c_i differing in a relative sense from those supplied by no more than $2.2 \times (\bar{n} + 3) \times$ ***machine precision***.

8 Further Comments

The time taken is approximately proportional to $\bar{n} + 7$.

9 Example

This example determines the definite integral over the interval $0 \leq x \leq 6$ of a cubic spline having 6 interior knots at the positions $\lambda =$ 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:

$\bar{n}$
LAMDA(j), for $j = 1, 2, \ldots,$ NCAP7
C(j), for $j = 1, 2, \ldots,$ NCAP7 $- 3$

9.1 Program Text

```
    Program e02bdfe

!     E02BDF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: e02bdf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: dint
      Integer                          :: ifail, ncap, ncap7
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: c(:), lamda(:)
!     .. Executable Statements ..
      Write (nout,*) 'E02BDF Example Program Results'

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

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

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

      ifail = 0
      Call e02bdf(ncap7,lamda,c,dint,ifail)

      Write (nout,*)
      Write (nout,99999) 'Definite integral = ', dint

99999 Format (1X,A,E11.3)
    End Program e02bdfe
```

9.2 Program Data

```
E02BDF Example Program Data
   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
```

9.3 Program Results

```
 E02BDF Example Program Results

 Definite integral =   0.100E+03
```