NAG Toolbox for MATLAB

e01ba

1 Purpose

e01ba determines a cubic spline interpolant to a given set of data.

2 Syntax

```
[lamda, c, ifail] = e01ba(x, y, 'm', m)
```

3 Description

e01ba 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 $x_1 < x_2 < \cdots < x_m$. 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 1975a):

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

where $N_i(x)$ denotes the normalized 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 function.

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; e01ba 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 Cox 1975a except that *QR* factorization is used instead of *LU* decomposition. The implementation of the algorithm involves setting up appropriate information for the related function e02ba followed by a call of that function. (See e02ba for further details.)

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

4 References

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

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) 627–668 Academic Press

5 Parameters

5.1 Compulsory Input Parameters

1: **x**(**m**) – **double array**

$\mathbf{x}(i)$ must be set to x_i, the ith data value of the independent variable x, for $i = 1, 2, \ldots, m$.

Constraint: $\mathbf{x}(i) < \mathbf{x}(i+1)$, for $i = 1, 2, \ldots, \mathbf{m} - 1$.

2: **y**(**m**) – **double array**

$\mathbf{y}(i)$ must be set to y_i, the ith data value of the dependent variable y, for $i = 1, 2, \ldots, m$.

5.2 Optional Input Parameters

1: **m – int32 scalar**

Default: The dimension of the arrays **x**, **y**. (An error is raised if these dimensions are not equal.)

m, the number of data points.

Constraint: $\mathbf{m} \geq 4$.

5.3 Input Parameters Omitted from the MATLAB Interface

lck, wrk, lwrk

5.4 Output Parameters

1: **lamda(lck) – double array**

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

2: **c(lck) – double array**

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.

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{m} < 4$,
or $\mathbf{lck} < \mathbf{m} + 4$,
or $\mathbf{lwrk} < 6 \times \mathbf{m} + 16$.

ifail $= 2$

The **x**-values fail to satisfy the condition

$\mathbf{x}(1) < \mathbf{x}(2) < \mathbf{x}(3) < \cdots < \mathbf{x}(\mathbf{m})$.

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 e01ba 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 1977) 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

```
[lamda, c, ifail] = e01ba(x, y, lck);
```

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

The value of $s(x)$ at $x = \mathbf{x}$ can be provided in the double variable **s** by the call

```
[s, ifail] = e02bb(lamda, c, x);
```

(see e02bb).

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

```
[s, ifail] = e02bc(lamda, c, x, left);
```

(see e02bc).

Here **left** must specify whether the left- or right-hand value of the third derivative is required (see e02bc for details).

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

```
[dint, ifail] = e02bd(lamda, c);
```

(see e02bd).

9 Example

```
x = [0;
     0.2;
     0.4;
     0.6;
     0.75;
     0.9;
     1];
y = [1;
     1.22140275816017;
     1.49182469764127;
     1.822118800390509;
     2.117000016612675;
     2.45960311115695;
     2.718281828459045];
[lamda, c, ifail] = e01ba(x, y)
```

```
lamda =
         0
         0
         0
         0
    0.4000
    0.6000
    0.7500
    1.0000
    1.0000
    1.0000
    1.0000
c =
    1.0000
    1.1336
    1.3726
    1.7827
    2.1744
    2.4918
    2.7183
         0
         0
         0
         0
ifail =
           0
```