# NAG Toolbox for MATLAB

# e01be

## 1 Purpose

e01be computes a monotonicity-preserving piecewise cubic Hermite interpolant to a set of data points.

## 2 Syntax

```
[d, ifail] = e01be(x, f, 'n', n)
```

## 3 Description

e01be estimates first derivatives at the set of data points $(x_r, f_r)$, for $r = 1, 2, \ldots, n$, which determine a piecewise cubic Hermite interpolant to the data, that preserves monotonicity over ranges where the data points are monotonic. If the data points are only piecewise monotonic, the interpolant will have an extremum at each point where monotonicity switches direction. The estimates of the derivatives are computed by a formula due to Brodlie, which is described in Fritsch and Butland 1984, with suitable changes at the boundary points.

The function is derived from function PCHIM in Fritsch 1982.

Values of the computed interpolant, and of its first derivative and definite integral, can subsequently be computed by calling e01bf, e01bg and e01bh, as described in Section 8.

## 4 References

Fritsch F N 1982 PCHIP final specifications *Report UCID-30194* Lawrence Livermore National Laboratory

Fritsch F N and Butland J 1984 A method for constructing local monotone piecewise cubic interpolants *SIAM J. Sci. Statist. Comput.* **5** 300–304

## 5 Parameters

### 5.1 Compulsory Input Parameters

1: **x**(**n**) – **double array**

**x**$(r)$ must be set to $x_r$, the $r$th value of the independent variable (abscissa), for $r = 1, 2, \ldots, n$.

*Constraint*: **x**$(r) <$ **x**$(r + 1)$.

2: **f**(**n**) – **double array**

**f**$(r)$ must be set to $f_r$, the $r$th value of the dependent variable (ordinate), for $r = 1, 2, \ldots, n$.

### 5.2 Optional Input Parameters

1: **n – int32 scalar**

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

$n$, the number of data points.

*Constraint*: **n** $\geq 2$.

### 5.3 Input Parameters Omitted from the MATLAB Interface

None.

### 5.4 Output Parameters

1: **d(n) – double array**

Estimates of derivatives at the data points. **d**($r$) contains the derivative at **x**($r$).

2: **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, **n** < 2.

**ifail** = 2

The values of **x**($r$), for $r = 1, 2, \ldots,$ **n**, are not in strictly increasing order.

## 7 Accuracy

The computational errors in the array **d** should be negligible in most practical situations.

## 8 Further Comments

The time taken by e01be is approximately proportional to $n$.

The values of the computed interpolant at the points **px**($i$), for $i = 1, 2, \ldots,$ **m**, may be obtained in the double array **pf**, of length at least **m**, by the call:

```
[pf, ifail] = e01bf(x, f, d, px);
```

where **n**, **x** and **f** are the input parameters to e01be and **d** is the output parameter from e01be.

The values of the computed interpolant at the points **px**($i$), for $i = 1, 2, \ldots,$ **m**, together with its first derivatives, may be obtained in the double arrays **pf** and **pd**, both of length at least **m**, by the call:

```
[pf, pd, ifail] = e01bg(x, f, d, px);
```

where **n**, **x**, **f** and **d** are as described above.

The value of the definite integral of the interpolant over the interval **a** to **b** can be obtained in the double variable **pint** by the call:

```
[pint, ifail] = e01bh(x, f, d, a, b);
```

where **n**, **x**, **f** and **d** are as described above.

## 9 Example

```
x = [7.99;
     8.09;
     8.19;
     8.699999999999999;
     9.199999999999999;
     10;
     12;
     15;
     20];
f = [0;
     2.7643e-05;
     0.04375;
```

```
     0.16918;
     0.46943;
     0.94374;
     0.99864;
     0.99992;
     0.99999];
[d, ifail] = e01be(x, f)
```

```
d =
         0
    0.0006
    0.3359
    0.3494
    0.5970
    0.0603
    0.0009
    0.0000
         0
ifail =
           0
```