

NAG Toolbox for MATLAB

e01bg

1 Purpose

e01bg evaluates a piecewise cubic Hermite interpolant and its first derivative at a set of points.

2 Syntax

```
[pf, pd, ifail] = e01bg(x, f, d, px, 'n', n, 'm', m)
```

3 Description

e01bg evaluates a piecewise cubic Hermite interpolant, as computed by e01be, at the points $\mathbf{px}(i)$, for $i = 1, 2, \dots, m$. The first derivatives at the points are also computed. If any point lies outside the interval from $\mathbf{x}(1)$ to $\mathbf{x}(n)$, values of the interpolant and its derivative are extrapolated from the nearest extreme cubic, and a warning is returned.

If values of the interpolant only, and not of its derivative, are required, e01bf should be used.

The function is derived from function PCHFD in Fritsch 1982.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

- 1: $\mathbf{x}(n)$ – double array
- 2: $\mathbf{f}(n)$ – double array
- 3: $\mathbf{d}(n)$ – double array

$\mathbf{n}$, $\mathbf{x}$, $\mathbf{f}$ and $\mathbf{d}$ must be unchanged from the previous call of e01be.

- 4: $\mathbf{px}(m)$ – double array

The m values of x at which the interpolant is to be evaluated.

5.2 Optional Input Parameters

- 1: $\mathbf{n}$ – int32 scalar

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

$\mathbf{n}$, $\mathbf{x}$, $\mathbf{f}$ and $\mathbf{d}$ must be unchanged from the previous call of e01be.

- 2: $\mathbf{m}$ – int32 scalar

Default: The dimension of the arrays $\mathbf{px}$, $\mathbf{pf}$, $\mathbf{pd}$. (An error is raised if these dimensions are not equal.)

m , the number of points at which the interpolant is to be evaluated.

Constraint: $\mathbf{m} \geq 1$.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

- 1: **pf(m) – double array**
pf(i) contains the value of the interpolant evaluated at the point **px(i)**, for $i = 1, 2, \dots, m$.
- 2: **pd(m) – double array**
pd(i) contains the first derivative of the interpolant evaluated at the point **px(i)**, for $i = 1, 2, \dots, m$.
- 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, **n** < 2.

ifail = 2

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

ifail = 3

On entry, **m** < 1.

ifail = 4

At least one of the points **px(i)**, for $i = 1, 2, \dots, \mathbf{m}$, lies outside the interval $[\mathbf{x}(1), \mathbf{x}(\mathbf{n})]$, and extrapolation was performed at all such points. Values computed at these points may be very unreliable.

7 Accuracy

The computational errors in the arrays **pf** and **pd** should be negligible in most practical situations.

8 Further Comments

The time taken by e01bg is approximately proportional to the number of evaluation points, m . The evaluation will be most efficient if the elements of **px** are in nondecreasing order (or, more generally, if they are grouped in increasing order of the intervals $[\mathbf{x}(r-1), \mathbf{x}(r)]$). A single call of e01bg with $m > 1$ is more efficient than several calls with $m = 1$.

9 Example

```
x = [7.99;
      8.09;
      8.19;
      8.699999999999999;
      9.199999999999999;
      10;
      12;
      15;
      20];
f = [0;
      2.7643e-05;
      0.043749;
```

```

    0.16918;
    0.46943;
    0.94374;
    0.99864;
    0.99992;
    0.99999];
d = [0;
    0.00055251;
    0.33587;
    0.34944;
    0.59696;
    0.060326;
    0.000898335;
    2.93954e-05;
    0];
px = [7.99;
    9.191000000000001;
    10.392;
    11.593;
    12.794;
    13.995;
    15.196;
    16.397;
    17.598;
    18.799;
    20];
[pf, pd, ifail] = e01bg(x, f, d, px)

pf =
    0
    0.4640
    0.9645
    0.9965
    0.9992
    0.9998
    0.9999
    1.0000
    1.0000
    1.0000
    1.0000
pd =
    0
    0.6060
    0.0457
    0.0099
    0.0006
    0.0003
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    -0.0000
ifail =
    0

```