NAG Toolbox

nag_interp_1d_monotonic (e01be)

1 Purpose

nag_interp_1d_monotonic (e01be) computes a monotonicity-preserving piecewise cubic Hermite interpolant to a set of data points.

2 Syntax

```
[d, ifail] = nag_interp_1d_monotonic(x, f, 'n', n)
[d, ifail] = e01be(x, f, 'n', n)
```

3 Description

nag_interp_1d_monotonic (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 nag_interp_1d_monotonic_eval (e01bf), nag_interp_1d_monotonic_deriv (e01bg) and nag_interp_1d_monotonic_intg (e01bh), as described in Section 9.

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**) – REAL (KIND=nag_wp) array

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

Constraint: $\mathbf{x}(r) < \mathbf{x}(r+1)$.

2: **f**(**n**) – REAL (KIND=nag_wp) array

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

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the number of data points.

Constraint: $\mathbf{n} \geq 2$.

5.3 Output Parameters

1: **d**(**n**) – REAL (KIND=nag_wp) array

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

2: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

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.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

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

8 Further Comments

The time taken by nag_interp_1d_monotonic (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 arguments to nag_interp_1d_monotonic (e01be) and **d** is the output argument from nag_interp_1d_monotonic (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

This example reads in a set of data points, calls nag_interp_1d_monotonic (e01be) to compute a piecewise monotonic interpolant, and then calls nag_interp_1d_monotonic_eval (e01bf) to evaluate the interpolant at equally spaced points.

9.1 Program Text

```
function e01be_example

fprintf('e01be example results\n\n');

x = [7.99 8.09    8.19    8.7     9.2     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);

m = 11;
dx = (x(end)-x(1))/(m-1);
px = [x(1):dx:x(end)];

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

fprintf('\n                Interpolated\n     Abscissa          Value\n');
fprintf('%13.4f  %13.4f\n', [px' pf]')
```

9.2 Program Results

```
e01be example results

           Interpolated
 Abscissa         Value
   7.9900        0.0000
   9.1910        0.4640
  10.3920        0.9645
  11.5930        0.9965
  12.7940        0.9992
  13.9950        0.9998
  15.1960        0.9999
  16.3970        1.0000
  17.5980        1.0000
  18.7990        1.0000
  20.0000        1.0000
```