NAG Toolbox

nag_interp_1d_monotonic_deriv (e01bg)

1 Purpose

nag_interp_1d_monotonic_deriv (e01bg) evaluates a piecewise cubic Hermite interpolant and its first derivative at a set of points.

2 Syntax

```
[pf, pd, ifail] = nag_interp_1d_monotonic_deriv(x, f, d, px, 'n', n, 'm', m)
[pf, pd, ifail] = e01bg(x, f, d, px, 'n', n, 'm', m)
```

3 Description

nag_interp_1d_monotonic_deriv (e01bg) evaluates a piecewise cubic Hermite interpolant, as computed by nag_interp_1d_monotonic (e01be), at the points **px**(i), for $i = 1, 2, \ldots, m$. The first derivatives at the points are also computed. If any point lies outside the interval from **x**(1) to **x**$(\mathbf{n})$, 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, nag_interp_1d_monotonic_eval (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: **x**(**n**) – REAL (KIND=nag_wp) array
2: **f**(**n**) – REAL (KIND=nag_wp) array
3: **d**(**n**) – REAL (KIND=nag_wp) array

n, **x**, **f** and **d** must be unchanged from the previous call of nag_interp_1d_monotonic (e01be).

4: **px**(**m**) – REAL (KIND=nag_wp) array

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

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, **x**, **f** and **d** must be unchanged from the previous call of nag_interp_1d_monotonic (e01be).

2: **m** – INTEGER

Default: the dimension of the array **px**.

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

Constraint: **m** ≥ 1.

5.3 Output Parameters

1: **pf**(**m**) – REAL (KIND=nag_wp) array

pf(i) contains the value of the interpolant evaluated at the point **px**(i), for $i = 1, 2, \ldots, m$.

2: **pd**(**m**) – REAL (KIND=nag_wp) array

pd(i) contains the first derivative of the interpolant evaluated at the point **px**(i), for $i = 1, 2, \ldots, m$.

3: **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 $= 3$

On entry, **m** < 1.

ifail $= 4$ (*warning*)

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

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 arrays **pf** and **pd** should be negligible in most practical situations.

8 Further Comments

The time taken by nag_interp_1d_monotonic_deriv (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 [**x**($r - 1$), **x**(r)]). A single call of nag_interp_1d_monotonic_deriv (e01bg) with $m > 1$ is more efficient than several calls with $m = 1$.

9 Example

This example reads in values of **n**, **x**, **f** and **d**, and calls nag_interp_1d_monotonic_deriv (e01bg) to compute the values of the interpolant and its derivative at equally spaced points.

9.1 Program Text

```
function e01bg_example

fprintf('e01bg 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];

% Theses are as returned by e01be(x,f)
d = [0;
     0.00055251;
     0.33587;
     0.34944;
     0.59696;
     0.060326;
     0.000898335;
     2.93954e-05;
     0];

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

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

fprintf('\n               Interpolated   Interpolated\n');
fprintf('     Abscissa          Value     Derivative\n');
fprintf('%13.4f%15.4f%15.3e\n', [px' pf pd]');

% Recalculate on finer mesh for plotting
m = 61;
dx = (x(end)-x(1))/(m-1);
px = [x(1):dx:x(end)];

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

fig1 = figure;
plot(px,pf,px,pd,x,f,'*');
legend('Function','Derivative','Data points','Location','East');
title('Monotonic Hermite interpolant');
xlabel('x');
axis([8 20 -0.1 1.2]);
```

9.2 Program Results

```
e01bg example results

           Interpolated   Interpolated
Abscissa          Value     Derivative
  7.9900         0.0000      0.000e+00
  9.1910         0.4640      6.060e-01
 10.3920         0.9645      4.569e-02
 11.5930         0.9965      9.917e-03
 12.7940         0.9992      6.249e-04
 13.9950         0.9998      2.708e-04
```

```
     15.1960          0.9999       2.809e-05
     16.3970          1.0000       2.034e-05
     17.5980          1.0000       1.308e-05
     18.7990          1.0000       6.297e-06
     20.0000          1.0000      -3.388e-21
```