NAG Toolbox

nag_interp_1d_everett (e01ab)

1 Purpose

nag_interp_1d_everett (e01ab) interpolates a function of one variable at a given point x from a table of function values evaluated at equidistant points, using Everett's formula.

2 Syntax

```
[a, g, ifail] = nag_interp_1d_everett(n, p, a)
[a, g, ifail] = e01ab(n, p, a)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 23: *n1* is no longer an optional input parameter; *n2* is no longer an input parameter.

3 Description

nag_interp_1d_everett (e01ab) interpolates a function of one variable at a given point

$$x = x_0 + ph,$$

where $-1 < p < 1$ and h is the interval of differencing, from a table of values $x_m = x_0 + mh$ and y_m where $m = -(n-1), -(n-2), \ldots, -1, 0, 1, \ldots, n$. The formula used is that of FrÎberg (1970), neglecting the remainder term:

$$y_p = \sum_{r=0}^{n-1} \left(\frac{1-p+r}{2r+1} \right) \delta^{2r} y_0 + \sum_{r=0}^{n-1} \left(\frac{p+r}{2r+1} \right) \delta^{2r} y_1.$$

The values of $\delta^{2r} y_0$ and $\delta^{2r} y_1$ are stored on exit from the function in addition to the interpolated function value y_p.

4 References

FrÎberg C E (1970) *Introduction to Numerical Analysis* Addison–Wesley

5 Parameters

5.1 Compulsory Input Parameters

1: **n** – INTEGER

n, half the number of points to be used in the interpolation.

Constraint: $\mathbf{n} > 0$.

2: **p** – REAL (KIND=nag_wp)

The point p at which the interpolated function value is required, i.e., $p = (x - x_0)/h$ with $-1.0 < p < 1.0$.

Constraint: $-1.0 < \mathbf{p} < 1.0$.

3: **a**($n1$) – REAL (KIND=nag_wp) array

a(i) must be set to the function value y_{i-n}, for $i = 1, 2, \ldots, 2n$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **a**$(n1)$ – REAL (KIND=nag_wp) array

$n1 = 2 \times$ **n**.

The contents of **a** are unspecified.

2: **g**$(n2)$ – REAL (KIND=nag_wp) array

$n2 = 2 \times$ **n** $+ 1$.

The array contains

y_0 in **g**(1)

y_1 in **g**(2)

$\delta^{2r} y_0$ in **g**$(2r+1)$

$\delta^{2r} y_1$ in **g**$(2r+2)$, for $r = 1, 2, \ldots, n-1$.

The interpolated function value y_p is stored in **g**$(2n+1)$.

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, **p** ≤ -1.0,
or **p** ≥ 1.0.

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

In general, increasing n improves the accuracy of the result until full attainable accuracy is reached, after which it might deteriorate. If x lies in the central interval of the data (i.e., $0.0 \leq p < 1.0$), as is desirable, an upper bound on the contribution of the highest order differences (which is usually an upper bound on the error of the result) is given approximately in terms of the elements of the array **g** by $a \times (|\mathbf{g}(2n-1)| + |\mathbf{g}(2n)|)$, where $a = 0.1,\ 0.02,\ 0.005,\ 0.001,\ 0.0002$ for $n = 1, 2, 3, 4, 5$ respectively, thereafter decreasing roughly by a factor of 4 each time.

8 Further Comments

The computation time increases as the order of n increases.

9 Example

This example interpolates at the point $x = 0.28$ from the function values

$$\begin{pmatrix} x_i & -1.00 & -0.50 & 0.00 & 0.50 & 1.00 & 1.50 \\ y_i & 0.00 & -0.53 & -1.00 & -0.46 & 2.00 & 11.09 \end{pmatrix}.$$

We take $n = 3$ and $p = 0.56$.

9.1 Program Text

```
     function e01ab_example

fprintf('e01ab example results\n\n');

a = [-1  -0.50   0    0.50    1    1.50];
b = [ 0  -0.53  -1   -0.46    2   11.09];
n = nag_int(size(a,2)/2);

x = 0.28;
% We get p from x = a(n) + p*h, where h = 0.5
p = (x-a(n))/0.5;

[bx, c, ifail] = e01ab(n, p, b);

for k = 0:n-1
  fprintf('Central differences order %4d of y_0 = %12.5f\n', k, c(2*k+1));
  fprintf('%37s = %12.5f\n', 'y_1',c(2*k+2));
end
fprintf('\nFunction value at interpolation point = %12.5f\n', c(end));
```

9.2 Program Results

```
     e01ab example results

Central differences order    0 of y_0 =     -1.00000
                                  y_1 =     -0.46000
Central differences order    1 of y_0 =      1.01000
                                  y_1 =      1.92000
Central differences order    2 of y_0 =     -0.04000
                                  y_1 =      3.80000

Function value at interpolation point =     -0.83591
```