NAG Toolbox

nag_interp_1d_aitken (e01aa)

1 Purpose

nag_interp_1d_aitken (e01aa) interpolates a function of one variable at a given point x from a table of function values y_i evaluated at equidistant or non-equidistant points x_i, for $i = 1, 2, \ldots, n + 1$, using Aitken's technique of successive linear interpolations.

2 Syntax

```
[a, b, c] = nag_interp_1d_aitken(a, b, n, x)
[a, b, c] = e01aa(a, b, n, x)
```

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_aitken (e01aa) interpolates a function of one variable at a given point x from a table of values x_i and y_i, for $i = 1, 2, \ldots, n + 1$ using Aitken's method (see FrÎberg (1970)). The intermediate values of linear interpolations are stored to enable an estimate of the accuracy of the results to be made.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

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

a(i) must contain the x-component of the ith data point, x_i, for $i = 1, 2, \ldots, n + 1$.

2: **b**(*n1*) – REAL (KIND=nag_wp) array

b(i) must contain the y-component (function value) of the ith data point, y_i, for $i = 1, 2, \ldots, n + 1$.

3: **n** – INTEGER

The number of intervals which are to be used in interpolating the value at x; that is, there are $n + 1$ data points (x_i, y_i).

Constraint: **n** > 0.

4: **x** – REAL (KIND=nag_wp)

The point x at which the interpolation is required.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

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

$n1 = \mathbf{n} + 1$.

a(i) contains the value $x_i - x$, for $i = 1, 2, \ldots, n + 1$.

2: **b**($n1$) – REAL (KIND=nag_wp) array

$n1 = \mathbf{n} + 1$.

The contents of **b** are unspecified.

3: **c**($n2$) – REAL (KIND=nag_wp) array

$n2 = \mathbf{n} \times (\mathbf{n} + 1)/2$.

$\mathbf{c}(1), \ldots, \mathbf{c}(n)$ contain the first set of linear interpolations,

$\mathbf{c}(n + 1), \ldots, \mathbf{c}(2 \times n - 1)$ contain the second set of linear interpolations,

$\mathbf{c}(2n), \ldots, \mathbf{c}(3 \times n - 3)$ contain the third set of linear interpolations,

$\vdots$

$\mathbf{c}(n \times (n + 1)/2)$ contains the interpolated function value at the point x.

6 Error Indicators and Warnings

None.

7 Accuracy

An estimate of the accuracy of the result can be made from a comparison of the final result and the previous interpolates, given in the array **c**. In particular, the first interpolate in the ith set, for $i = 1, 2, \ldots, n$, is the value at x of the polynomial interpolating the first $(i + 1)$ data points. It is given in position $(i - 1)(2n - i + 2)/2$ of the array **c**. Ideally, providing n is large enough, this set of n interpolates should exhibit convergence to the final value, the difference between one interpolate and the next settling down to a roughly constant magnitude (but with varying sign). This magnitude indicates the size of the error (any subsequent increase meaning that the value of n is too high). Better convergence will be obtained if the data points are supplied, not in their natural order, but ordered so that the first i data points give good coverage of the neighbourhood of x, for all i. To this end, the following ordering is recommended as widely suitable: first the point nearest to x, then the nearest point on the opposite side of x, followed by the remaining points in increasing order of their distance from x, that is of $|x_r - x|$. With this modification the Aitken method will generally perform better than the related method of Neville, which is often given in the literature as superior to that of Aitken.

8 Further Comments

The computation time for interpolation at any point x is proportional to $n \times (n + 1)/2$.

9 Example

This example interpolates at $x = 0.28$ the function value of a curve defined by the points

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

9.1 Program Text

```
function e01aa_example

fprintf('e01aa 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(5);

x = 0.28;
[ax, bx, c] = e01aa(a, b, n, x);

k = 1;

disp('Interpolated values');
for i = n-1:-1:1
  fprintf('%12.5f',c(k:k+i));
  fprintf('\n');
  k = k + i + 1;
end

fprintf('\nInterpolation point = %12.5f\n', x);
fprintf('\nFunction value at interpolation point = %12.5f\n', c(end));
```

9.2 Program Results

```
e01aa example results

Interpolated values
    -1.35680    -1.28000    -0.39253     1.28000     5.67808
    -1.23699    -0.60467     0.01434     1.38680
    -0.88289    -0.88662    -0.74722
    -0.88125    -0.91274

Interpolation point =      0.28000

Function value at interpolation point =     -0.83591
```