NAG Toolbox for MATLAB

e01aa

1 Purpose

e01aa interpolates 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] = e01aa(a, b, n2, n, x, 'n1', n1)
```

3 Description

e01aa interpolates 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. The intermediate values of linear interpolations are stored to enable an estimate of the accuracy of the results to be made.

4 References

Fröberg C E 1970 *Introduction to Numerical Analysis* Addison–Wesley

5 Parameters

5.1 Compulsory Input Parameters

1: **a(n1) – double 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) – double 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: **n2 – int32 scalar**

the value $n \times (n + 1)/2$ where n is the number of intervals.

4: **n – int32 scalar**

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).

5: **x – double scalar**

The point x at which the interpolation is required.

5.2 Optional Input Parameters

1: **n1 – int32 scalar**

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

the value $n + 1$ where n is the number of intervals; that is, **n1** is the number of data points.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **a(n1) – double array**

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

2: **b(n1) – double array**

The contents of **b** are unspecified.

3: **c(n2) – double array**

$\mathbf{c}(1), \ldots, \mathbf{c}(n)$ 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

$\vdots$

$\mathbf{c}(n \times (n + 1)/2)$ 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 $1 + \frac{1}{2}(i - 1)(2n - i + 2)$ 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

```
a = [-1;
     -0.5;
     0;
     0.5;
     1;
     1.5];
b = [0;
     -0.53;
     -1;
     -0.46;
     2;
     11.09];
n2 = int32(15);
n = int32(5);
```

```
x = 0.28;
[aOut, bOut, c] = e01aa(a, b, n2, n, x)
```

```
aOut =
   -1.2800
   -0.7800
   -0.2800
    0.2200
    0.7200
    1.2200
bOut =
         0
   -1.3568
   -1.2370
   -0.8829
   -0.8813
   -0.8359
c =
   -1.3568
   -1.2800
   -0.3925
    1.2800
    5.6781
   -1.2370
   -0.6047
    0.0143
    1.3868
   -0.8829
   -0.8866
   -0.7472
   -0.8813
   -0.9127
   -0.8359
```