NAG Toolbox for MATLAB

e02ah

1 Purpose

e02ah determines the coefficients in the Chebyshev-series representation of the derivative of a polynomial given in Chebyshev-series form.

2 Syntax

```
[patm1, adif, ifail] = e02ah(n, xmin, xmax, a, ia1, iadif1)
```

3 Description

e02ah forms the polynomial which is the derivative of a given polynomial. Both the original polynomial and its derivative are represented in Chebyshev-series form. Given the coefficients a_i, for $i = 0, 1, \ldots, n$, of a polynomial $p(x)$ of degree n, where

$$p(x) = \tfrac{1}{2}a_0 + a_1 T_1(\bar{x}) + \cdots + a_n T_n(\bar{x})$$

the function returns the coefficients $\bar{a}_i$, for $i = 0, 1, \ldots, n-1$, of the polynomial $q(x)$ of degree $n-1$, where

$$q(x) = \frac{dp(x)}{dx} = \tfrac{1}{2}\bar{a}_0 + \bar{a}_1 T_1(\bar{x}) + \cdots + \bar{a}_{n-1} T_{n-1}(\bar{x}).$$

Here $T_j(\bar{x})$ denotes the Chebyshev polynomial of the first kind of degree j with argument $\bar{x}$. It is assumed that the normalized variable $\bar{x}$ in the interval $[-1, +1]$ was obtained from your original variable x in the interval $[x_{\mathrm{min}}, x_{\mathrm{max}}]$ by the linear transformation

$$\bar{x} = \frac{2x - (x_{\mathrm{max}} + x_{\mathrm{min}})}{x_{\mathrm{max}} - x_{\mathrm{min}}}$$

and that you require the derivative to be with respect to the variable x. If the derivative with respect to $\bar{x}$ is required, set $x_{\mathrm{max}} = 1$ and $x_{\mathrm{min}} = -1$.

Values of the derivative can subsequently be computed, from the coefficients obtained, by using e02ak.

The method employed is that of Chebyshev-series (see Chapter 8 of Modern Computing Methods 1961), modified to obtain the derivative with respect to x. Initially setting $\bar{a}_{n+1} = \bar{a}_n = 0$, the function forms successively

$$\bar{a}_{i-1} = \bar{a}_{i+1} + \frac{2}{x_{\mathrm{max}} - x_{\mathrm{min}}} 2ia_i, \qquad i = n, n-1, \ldots, 1.$$

4 References

Modern Computing Methods 1961 Chebyshev-series *NPL Notes on Applied Science* **16** (2nd Edition) HMSO

5 Parameters

5.1 Compulsory Input Parameters

1: **n – int32 scalar**

n, the degree of the given polynomial $p(x)$.

Constraint: **n** ≥ 0.

2: **xmin – double scalar**

3: **xmax – double scalar**

The lower and upper end points respectively of the interval $[x_{\min}, x_{\max}]$. The Chebyshev-series representation is in terms of the normalized variable $\bar{x}$, where

$$\bar{x} = \frac{2x - (x_{\max} + x_{\min})}{x_{\max} - x_{\min}}.$$

Constraint: **xmax** $>$ **xmin**.

4: **a(la) – double array**

The Chebyshev coefficients of the polynomial $p(x)$. Specifically, element $i \times$ **ia1** of **a** must contain the coefficient a_i, for $i = 0, 1, \ldots, n$. Only these $n + 1$ elements will be accessed.

Unchanged on exit, but see **adif**, below.

5: **ia1 – int32 scalar**

The index increment of **a**. Most frequently the Chebyshev coefficients are stored in adjacent elements of **a**, and **ia1** must be set to 1. However, if, for example, they are stored in **a**(1), **a**(4), **a**(7), ..., then the value of **ia1** must be 3. See also Section 8.

Constraint: **ia1** ≥ 1.

6: **iadif1 – int32 scalar**

The index increment of **adif**. Most frequently the Chebyshev coefficients are required in adjacent elements of **adif**, and **iadif1** must be set to 1. However, if, for example, they are to be stored in **adif**(1), **adif**(4), **adif**(7), ..., then the value of **iadif1** must be 3. See Section 8.

Constraint: **iadif1** ≥ 1.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

np1, la, ladif

5.4 Output Parameters

1: **patm1 – double scalar**

The value of $p(x_{\min})$. If this value is passed to the integration function e02aj with the coefficients of $q(x)$, then the original polynomial $p(x)$ is recovered, including its constant coefficient.

2: **adif(ladif) – double array**

The Chebyshev coefficients of the derived polynomial $q(x)$. (The differentiation is with respect to the variable x.) Specifically, element $i \times$ **iadif1** $+ 1$ of **adif** contains the coefficient $\bar{a}_i$, $i = 0, 1, \ldots, n - 1$. Additionally, element $n \times$ **iadif1** $+ 1$ is set to zero. A call of the function may have the array name **adif** the same as **a**, provided that note is taken of the order in which elements are overwritten, when choosing the starting elements and increments **ia1** and **iadif1**: i.e., the coefficients $a_0, a_1, \ldots, a_{i-1}$ must be intact after coefficient $\bar{a}_i$ is stored. In particular, it is possible to overwrite the a_i completely by having **ia1** $=$ **iadif1**, and the actual arrays for **a** and **adif** identical.

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **np1** < 1,
or **xmax** $\leq$ **xmin**,
or **ia1** < 1,
or **la** $\leq$ (**np1** $- 1) \times$ **ia1**,
or **iadif1** < 1,
or **ladif** $\leq$ (**np1** $- 1) \times$ **iadif1**.

7 Accuracy

There is always a loss of precision in numerical differentiation, in this case associated with the multiplication by $2i$ in the formula quoted in Section 3.

8 Further Comments

The time taken is approximately proportional to $n + 1$.

The increments **ia1**, **iadif1** are included as parameters to give a degree of flexibility which, for example, allows a polynomial in two variables to be differentiated with respect to either variable without rearranging the coefficients.

9 Example

```
n = int32(6);
xmin = -0.5;
xmax = 2.5;
a = [2.53213;
     1.13032;
     0.2715;
     0.04434;
     0.00547;
     0.00054;
     4e-05];
ia1 = int32(1);
iadif1 = int32(1);
[patm1, adif, ifail] = e02ah(n, xmin, xmax, a, ia1, iadif1)
```

```
patm1 =
    0.3679
adif =
    1.6881
    0.7535
    0.1810
    0.0295
    0.0036
    0.0003
         0
ifail =
           0
```