

NAG Fortran Library Routine Document

E02AHF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

E02AHF determines the coefficients in the Chebyshev-series representation of the derivative of a polynomial given in Chebyshev-series form.

2 Specification

```

SUBROUTINE E02AHF(NP1, XMIN, XMAX, A, IA1, LA, PATM1, ADIF, IADIF1,
1                LADIF, IFAIL)
  INTEGER          NP1, IA1, LA, IADIF1, LADIF, IFAIL
  real           XMIN, XMAX, A(LA), PATM1, ADIF(LADIF)

```

3 Description

This routine 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 $i = 0, 1, \dots, n$, of a polynomial $p(x)$ of degree n , where

$$p(x) = \frac{1}{2}a_0 + a_1T_1(\bar{x}) + \dots + a_nT_n(\bar{x})$$

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

$$q(x) = \frac{dp(x)}{dx} = \frac{1}{2}\bar{a}_0 + \bar{a}_1T_1(\bar{x}) + \dots + \bar{a}_{n-1}T_{n-1}(\bar{x}).$$

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 normalised variable \bar{x} in the interval $[-1, +1]$ was obtained from the user's original variable x in the interval $[x_{\min}, x_{\max}]$ by the linear transformation

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

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

Values of the derivative can subsequently be computed, from the coefficients obtained, by using E02AKF.

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

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

4 References

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

5 Parameters

1: NP1 – INTEGER *Input*

On entry: $n + 1$, where n is the degree of the given polynomial $p(x)$. Thus NP1 is the number of coefficients in this polynomial.

Constraint: $NP1 \geq 1$.

2: XMIN – *real* *Input*

3: XMAX – *real* *Input*

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

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

Constraint: $XMAX > XMIN$.

4: A(LA) – *real* array *Input*

On entry: the Chebyshev coefficients of the polynomial $p(x)$. Specifically, element $1 + i \times IA1$ of A must contain the coefficient a_i , for $i = 0, 1, \dots, n$. Only these $n + 1$ elements will be accessed.

Unchanged on exit, but see ADIF, below.

5: IA1 – INTEGER *Input*

On entry: 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 $A(1), A(4), A(7), \dots$, then the value of IA1 must be 3. See also Section 8.

Constraint: $IA1 \geq 1$.

6: LA – INTEGER *Input*

On entry: the dimension of the array A as declared in the (sub)program from which E02AHF is called.

Constraint: $LA \geq 1 + (NP1 - 1) \times IA1$.

7: PATM1 – *real* *Output*

On exit: the value of $p(x_{\min})$. If this value is passed to the integration routine E02AJF with the coefficients of $q(x)$, then the original polynomial $p(x)$ is recovered, including its constant coefficient.

8: ADIF(LADIF) – *real* array *Output*

On exit: the Chebyshev coefficients of the derived polynomial $q(x)$. (The differentiation is with respect to the variable x .) Specifically, element $1 + i \times IADIF1$ of ADIF contains the coefficient \bar{a}_i , $i = 0, 1, \dots, n - 1$. Additionally element $1 + n \times IADIF1$ is set to zero. A call of the routine 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, \dots, 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.

9: IADIF1 – INTEGER *Input*

On entry: 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 $ADIF(1), ADIF(4), ADIF(7), \dots$, then the value of IADIF1 must be 3. See Section 8.

Constraint: $IADIF1 \geq 1$.

10: LADIF – INTEGER

Input

On entry: the dimension of the array ADIF as declared in the (sub)program from which E02AHF is called.

Constraint: $LADIF \geq 1 + (NP1 - 1) \times IADIF1$.

11: IFAIL – INTEGER

Input/Output

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

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 by the routine 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

Suppose a polynomial has been computed in Chebyshev-series form to fit data over the interval $[-0.5, 2.5]$. The following program evaluates the first and second derivatives of this polynomial at 4 equally spaced points over the interval. (For the purposes of this example, XMIN, XMAX and the Chebyshev coefficients are simply supplied in DATA statements. Normally a program would first read in or generate data and compute the fitted polynomial.)

9.1 Program Text

Note: the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```

*      E02AHF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
INTEGER      NP1, LA, LADIF
PARAMETER    (NP1=7,LA=NP1,LADIF=NP1)
INTEGER      NOUT
PARAMETER    (NOUT=6)
*      .. Local Scalars ..
real        DERIV, DERIV2, PATM1, X, XMAX, XMIN
INTEGER      I, IFAIL, M
*      .. Local Arrays ..
real        A(LA), ADIF(LADIF), ADIF2(LADIF)
*      .. External Subroutines ..
EXTERNAL     E02AHF, E02AKF
*      .. Intrinsic Functions ..
INTRINSIC    real
*      .. Data statements ..
DATA        XMIN, XMAX/-0.5e0, 2.5e0/
DATA        (A(I),I=1,NP1)/2.53213e0, 1.13032e0, 0.27150e0,
+           0.04434e0, 0.00547e0, 0.00054e0, 0.00004e0/
*      .. Executable Statements ..
WRITE (NOUT,*) 'E02AHF Example Program Results'
IFAIL = 0
*
CALL E02AHF(NP1,XMIN,XMAX,A,1,LA,PATM1,ADIF,1,LADIF,IFAIL)
CALL E02AHF(NP1-1,XMIN,XMAX,ADIF,1,LADIF,PATM1,ADIF2,1,LADIF,
+           IFAIL)
*
M = 4
WRITE (NOUT,*)
WRITE (NOUT,*) '  I Argument      1st deriv      2nd deriv'
DO 20 I = 1, M
  X = (XMIN*real(M-I)+XMAX*real(I-1))/real(M-1)
*
  CALL E02AKF(NP1-1,XMIN,XMAX,ADIF,1,LADIF,X,DERIV,IFAIL)
  CALL E02AKF(NP1-2,XMIN,XMAX,ADIF2,1,LADIF,X,DERIV2,IFAIL)
*
  WRITE (NOUT,99999) I, X, DERIV, DERIV2
20 CONTINUE
STOP
*
99999 FORMAT (1X,I4,F9.4,2(4X,F9.4))
END

```

9.2 Program Data

None.

9.3 Program Results

E02AHF Example Program Results

I	Argument	1st deriv	2nd deriv
1	-0.5000	0.2453	0.1637
2	0.5000	0.4777	0.3185
3	1.5000	0.9304	0.6203
4	2.5000	1.8119	1.2056