

NAG Fortran Library Routine Document

E02AEF

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

E02AEF evaluates a polynomial from its Chebyshev-series representation.

2 Specification

```
SUBROUTINE E02AEF(NPLUS1, A, XCAP, P, IFAIL)
INTEGER          NPLUS1, IFAIL
real           A(NPLUS1), XCAP, P
```

3 Description

This routine evaluates the polynomial

$$\frac{1}{2}a_1T_0(\bar{x}) + a_2T_1(\bar{x}) + a_3T_2(\bar{x}) + \cdots + a_{n+1}T_n(\bar{x})$$

for any value of \bar{x} satisfying $-1 \leq \bar{x} \leq 1$. Here $T_j(\bar{x})$ denotes the Chebyshev polynomial of the first kind of degree j with argument \bar{x} . The value of n is prescribed by the user.

In practice, the variable \bar{x} will usually have been obtained from an original variable x , where $x_{\min} \leq x \leq x_{\max}$ and

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

Note that this form of the transformation should be used computationally rather than the mathematical equivalent

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

since the former guarantees that the computed value of \bar{x} differs from its true value by at most 4ϵ , where ϵ is the ***machine precision***, whereas the latter has no such guarantee.

The method employed is based upon the three-term recurrence relation due to Clenshaw (1955), with modifications to give greater numerical stability due to Reinsch and Gentleman (see Gentleman (1969)).

For further details of the algorithm and its use see Cox (1974) and Cox and Hayes (1973).

4 References

Clenshaw C W (1955) A note on the summation of Chebyshev-series *Math. Tables Aids Comput.* **9** 118–120

Cox M G (1974) A data-fitting package for the non-specialist user *Software for Numerical Mathematics* (ed D J Evans) Academic Press

Cox M G and Hayes J G (1973) Curve fitting: a guide and suite of algorithms for the non-specialist user *NPL Report NAC26* National Physical Laboratory

Gentleman W M (1969) An error analysis of Goertzel's (Watt's) method for computing Fourier coefficients *Comput. J.* **12** 160–165

5 Parameters

- 1: NPLUS1 – INTEGER *Input*
On entry: the number $n + 1$ of terms in the series (i.e., one greater than the degree of the polynomial).
Constraint: $NPLUS1 \geq 1$.
- 2: A(NPLUS1) – *real* array *Input*
On entry: $A(i)$ must be set to the value of the i th coefficient in the series, for $i = 1, 2, \dots, n + 1$.
- 3: XCAP – *real* *Input*
On entry: \bar{x} , the argument at which the polynomial is to be evaluated. It should lie in the range -1 to $+1$, but a value just outside this range is permitted (see Section 6) to allow for possible rounding errors committed in the transformation from x to \bar{x} discussed in Section 3. Provided the recommended form of the transformation is used, a successful exit is thus assured whenever the value of x lies in the range x_{\min} to x_{\max} .
- 4: P – *real* *Output*
On exit: the value of the polynomial.
- 5: 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:

$IFAIL = 1$

$ABS(XCAP) > 1.0 + 4\epsilon$, where ϵ is the *machine precision*. In this case the value of P is set arbitrarily to zero.

$IFAIL = 2$

On entry, $NPLUS1 < 1$.

7 Accuracy

The rounding errors committed are such that the computed value of the polynomial is exact for a slightly perturbed set of coefficients $a_i + \delta a_i$. The ratio of the sum of the absolute values of the δa_i to the sum of the absolute values of the a_i is less than a small multiple of $(n + 1)$ times *machine precision*.

8 Further Comments

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

It is expected that a common use of E02AEF will be the evaluation of the polynomial approximations produced by E02ADF and E02AFF.

9 Example

Evaluate at 11 equally-spaced points in the interval $-1 \leq \bar{x} \leq 1$ the polynomial of degree 4 with Chebyshev coefficients, 2.0, 0.5, 0.25, 0.125, 0.0625.

The example program is written in a general form that will enable a polynomial of degree n in its Chebyshev-series form to be evaluated at m equally-spaced points in the interval $-1 \leq \bar{x} \leq 1$. The program is self-starting in that any number of data sets can be supplied.

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.

```
*      E02AEF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          NMAX, NP1MAX
      PARAMETER       (NMAX=199, NP1MAX=NMAX+1)
      INTEGER          NIN, NOUT
      PARAMETER       (NIN=5, NOUT=6)
*      .. Local Scalars ..
      real            P, XCAP
      INTEGER          I, IFAIL, M, N, R
*      .. Local Arrays ..
      real            A(NP1MAX)
*      .. External Subroutines ..
      EXTERNAL        E02AEF
*      .. Intrinsic Functions ..
      INTRINSIC       real
*      .. Executable Statements ..
      WRITE (NOUT,*) 'E02AEF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
20  READ (NIN,*,END=60) M
      IF (M.GT.0) THEN
          READ (NIN,*) N
          IF (N.GE.0 .AND. N.LE.NMAX) THEN
              READ (NIN,*) (A(I),I=1,N+1)
              WRITE (NOUT,*)
              WRITE (NOUT,*)
+          ' R      Argument      Value of polynomial'
              DO 40 R = 1, M
                  XCAP = real(2*R-M-1)/real(M-1)
                  IFAIL = 0
*
*                  CALL E02AEF(N+1,A,XCAP,P,IFAIL)
*
              WRITE (NOUT,99999) R, XCAP, P
40      CONTINUE
              GO TO 20
          END IF
      END IF
60  STOP
*
99999 FORMAT (1X,I3,F14.4,4X,F14.4)
      END
```

9.2 Program Data

E02AEF Example Program Data

```
11
 4
 2.0000
 0.5000
 0.2500
 0.1250
 0.0625
```

9.3 Program Results

E02AEF Example Program Results

R	Argument	Value of polynomial
1	-1.0000	0.6875
2	-0.8000	0.6613
3	-0.6000	0.6943
4	-0.4000	0.7433
5	-0.2000	0.7843
6	0.0000	0.8125
7	0.2000	0.8423
8	0.4000	0.9073
9	0.6000	1.0603
10	0.8000	1.3733
11	1.0000	1.9375
