NAG Fortran Library Routine Document

E02RBF

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

E02RBF evaluates a rational function at a user-supplied point, given the numerator and denominator coefficients.

2 Specification

```
SUBROUTINE E02RBF(A, IA, B, IB, X, ANS, IFAIL)
INTEGER            IA, IB, IFAIL
real               A(IA), B(IB), X, ANS
```

3 Description

Given a real value x and the coefficients a_j, for $j = 0, \ldots, l$ and b_k, for $k = 0, \ldots, m$, E02RBF evaluates the rational function

$$\frac{\sum_{j=0}^{l} a_j x^j}{\sum_{k=0}^{m} b_k x^k}.$$

using nested multiplication (Conte and de Boor (1965)).

A particular use of E02RBF is to compute values of the Padé approximants determined by E02RAF.

4 References

Conte S D and de Boor C (1965) *Elementary Numerical Analysis* McGraw-Hill

Peters G and Wilkinson J H (1971) Practical problems arising in the solution of polynomial equations *J. Inst. Maths. Applics.* **8** 16–35

5 Parameters

1: A(IA) – ***real*** array *Input*

On entry: A$(j+1)$, for $j = 1, 2, \ldots, l+1$, must contain the value of the coefficient a_j in the numerator of the rational function.

2: IA – INTEGER *Input*

On entry: the value of $l+1$, where l is the degree of the numerator.

Constraint: IA ≥ 1.

3: B(IB) – ***real*** array *Input*

On entry: B$(k+1)$, for $k = 1, 2, \ldots, m+1$, must contain the value of the coefficient b_k in the denominator of the rational function.

Constraint: if IB $= 1$, B(1) must be non-zero.

4: IB – INTEGER *Input*

On entry: the value of $m + 1$, where m is the degree of the denominator.

Constraint: IB ≥ 1.

5: X – ***real*** *Input*

On entry: the point x at which the rational function is to be evaluated.

6: ANS – ***real*** *Output*

On exit: the result of evaluating the rational function at the given point x.

7: 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 $= 1$

The rational function is being evaluated at or near a pole.

IFAIL $= 2$

On entry, IA < 1,
or IB < 1,
or B$(1) = 0.0$ when IB $= 1$ (so the denominator is identically zero).

7 Accuracy

A running error analysis for polynomial evaluation by nested multiplication using the recurrence suggested by Kahan (see Peters and Wilkinson (1971)) is used to detect whether the user is attempting to evaluate the approximant at or near a pole.

8 Further Comments

The time taken by the routine is approximately proportional to $l + m$.

9 Example

The example program first calls E02RAF to calculate the 4/4 Padé approximant to e^x, and then uses E02RBF to evaluate the approximant at $x = 0.1, 0.2, \ldots, 1.0$.

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.

```
*     E02RBF Example Program Text.
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          L, M, IA, IB, IC, IW
      PARAMETER        (L=4,M=4,IA=L+1,IB=M+1,IC=IA+IB-1,IW=IB*(2*IB+3))
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             ANS, TVAL, X
      INTEGER          I, IFAIL
*     .. Local Arrays ..
      real             AA(IA), BB(IB), CC(IC), W(IW)
*     .. External Subroutines ..
      EXTERNAL         E02RAF, E02RBF
*     .. Intrinsic Functions ..
      INTRINSIC        EXP, real
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E02RBF Example Program Results'
      CC(1) = 1.0e0
      DO 20 I = 1, IC - 1
         CC(I+1) = CC(I)/real(I)
   20 CONTINUE
      IFAIL = 0
*
      CALL E02RAF(IA,IB,CC,IC,AA,BB,W,IW,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) '    X        Pade           True'
      DO 40 I = 1, 10
         X = real(I)/10.0e0
         IFAIL = 0
*
         CALL E02RBF(AA,IA,BB,IB,X,ANS,IFAIL)
*
         TVAL = EXP(X)
         WRITE (NOUT,99999) X, ANS, TVAL
   40 CONTINUE
      STOP
*
99999 FORMAT (1X,F6.1,3e15.5)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 E02RBF Example Program Results

     X        Pade           True
    0.1    0.11052E+01    0.11052E+01
    0.2    0.12214E+01    0.12214E+01
    0.3    0.13499E+01    0.13499E+01
    0.4    0.14918E+01    0.14918E+01
    0.5    0.16487E+01    0.16487E+01
    0.6    0.18221E+01    0.18221E+01
    0.7    0.20138E+01    0.20138E+01
    0.8    0.22255E+01    0.22255E+01
    0.9    0.24596E+01    0.24596E+01
    1.0    0.27183E+01    0.27183E+01
```